

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ALEXANDRE SANTOS BRANDÃO

CONTROLE DESCENTRALIZADO COM DESVIO DE OBSTÁCULOS
PARA UMA FORMAÇÃO LÍDER-SEGUIDOR DE ROBÔS MÓVEIS

VITÓRIA - ES, BRASIL
MARÇO DE 2008

ALEXANDRE SANTOS BRANDÃO

**CONTROLE DESCENTRALIZADO COM DESVIO DE OBSTÁCULOS
PARA UMA FORMAÇÃO LÍDER-SEGUIDOR DE ROBÔS MÓVEIS**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Engenharia Elétrica, na área de concentração em Automação.

Orientador: Prof. Dr. Mário Sarcinelli Filho

VITÓRIA, ES - BRASIL

MARÇO DE 2008

Dados Internacionais de Catalogação-na-publicação (CIP)
(Biblioteca Central da Universidade Federal do Espírito Santo, ES, Brasil)

Brandão, Alexandre Santos, 1982-
B817c Controle descentralizado com desvio de obstáculos para uma
 formação líder-seguidor de robôs móveis / Alexandre Santos Brandão. -
 2008.
 120 f. : il.

Orientador: Mário Sarcinelli Filho.
Co-Orientadores: Teodiano Freire Bastos Filho, Ricardo Carelli.
Dissertação (mestrado) - Universidade Federal do Espírito Santo,
Centro Tecnológico.

1. Robótica. 2. Robôs móveis. 3. Desvio de obstáculos. 4. Lasers. 5.
Distâncias - Medição. I. Sarcinelli Filho, Mário. II. Bastos Filho, Teodiano
Freire. III. Carelli, Ricardo. IV. Universidade Federal do Espírito Santo.
Centro Tecnológico. V. Título.

CDU: 621.3

ALEXANDRE SANTOS BRANDÃO

**CONTROLE DESCENTRALIZADO COM DESVIO DE OBSTÁCULOS
PARA UMA FORMAÇÃO LÍDER-SEGUIDOR DE ROBÔS MÓVEIS**

Dissertação submetida ao programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisição parcial para a obtenção do Grau de Mestre em Engenharia Elétrica - Automação.

Aprovada em 07 de março de 2008.

COMISSÃO EXAMINADORA

Prof. Dr. Mário Sarcinelli Filho
Universidade Federal do Espírito Santo
Orientador

Prof. Dr. Teodiano Freire Bastos Filho
Universidade Federal do Espírito Santo
Co-orientador

Prof. Dr. Ricardo Carelli
Universidad Nacional de San Juan - Argentina
Co-orientador

Prof. Dr. Paulo Faria Santos Amaral
Universidade Federal do Espírito Santo

Prof. Dr. Jês de Jesus Fiais Cerqueira
Universidade Federal da Bahia

*Dedico este trabalho às mulheres de minha vida:
Cecília (mãe), Celina (tia), Simone (irmã) e Zilma (avó - in memoriam).*

Agradecimentos

Ao Prof. Dr. Mário Sarcinelli Filho, meu orientador, eu dedico meus leais agradecimentos pelo norteamento e pela confiança em mim depositada. Por acreditar fielmente no trabalho desenvolvido e em seus resultados de êxito, a ponto de me conceder a oportunidade de abordar cada item desta Dissertação de duas formas distintas. Sou grato também às suas frases de incentivo moral, em especial aquela dita no instante em que ingressei no Programa de Pós-Graduação: “*Você irá trabalhar no Laboratório de Automação **Inteligente!** Sente-se capaz para isto?*”. Espero sinceramente, *My Professor*, ter atendido suas expectativas.

Ao Prof. Dr. Teodiano Freire Bastos Filho, meu co-orientador, agradeço por me auxiliar nos primeiros passos durante a familiarização com a Robótica. Desejo também demonstrar minha gratidão pelos momentos de discussão, sempre disputados por outros de seus orientados. E, por fim, gostaria de agradecer às revisões precisas de meus *artículos* escritos em espanhol.

Ao Prof. Dr. Ricardo Carelli, meu co-orientador à distância, sempre presente, sou extremamente grato por me receber de braços abertos no Instituto de Automática durante as minhas estadas em San Juan - Argentina. Assim como por dedicar parte de seu precioso tempo às reuniões técnicas essenciais para a execução do meu trabalho, fornecendo a mim muito mais que simples palavras de apoio e sugestões primordiais.

Ao CNPq, por me conceder a bolsa de estudos, que possibilitou minha subsistência durante a execução do meu mestrado.

À CAPES, por viabilizar financeiramente minhas estadas em San Juan - Argentina.

Ao meu irmão siamês, Daniel Cruz Cavalieri, sou grato pelos anos de companheirismo compartilhados dentro e fora do ambiente universitário. Agradeço, novamente, por todas às vezes que parou seu trabalho para me ouvir comentar sobre o meu, e, mesmo sem entender, prestava atenção em minhas palavras de forma serena.

Ao Wanderley Cardoso e ao Felipe Martins, pelos momentos inesquecíveis vividos na SQSJ, durante nossa realização do intercâmbio na Argentina. Dentre os quais destaco as noites sem dormir antes da prova de Controle de Robôs e nosso “turismo científico”.

À Cynthia Netto, por seu poder de persuasão, essencial para o desenvolvimento deste trabalho durante as estadas no INAUT.

Aos companheiros de trabalho do LAI1 e LAI2, em especial, André Ferreira, Flávio Garcia e Christiano Gava, pela paciência e dedicação em sanar minhas dúvidas sobre o Pioneer.

Aos colegas, professores e funcionários do INAUT/UNSJ, em especial, Jose Luiz, Franco Penizzotto, Lorgio Teodovich, Carlos Soria e Fernando Auat, que conviveram no laboratório nomeado pelos brasileiros de LAI3.

Aos amigos de graduação na UFV, Eduardo Donato e Filipe Lima, pelos momentos de descontração propiciados pelo PES6.

A todos os demais professores do PPGEE/UFES, agradeço pela força e sabedoria transmitidas durante minha formação.

A todos os amigos e colegas de diferentes cidades, estados, e até mesmo países, conhecidos durante meu Mestrado.

Gostaria de agradecer profundamente aos meus pais, Cecília dos Santos e Oséas Brandão, à minha tia Celina dos Santos e à minha irmã Simone Brandão, por me amar, apoiar, opinar, escutar os meus lamentos e, acima de tudo, incentivar à conclusão de mais uma etapa da minha formação científica e, principalmente, pessoal; mesmo estando a vários quilômetros de distância, em minha querida cidade de Timóteo/MG. Ao meu sobrinho Vitor Brandão Cordeiro, gostaria de agradecer pela força expressa em forma de sorrisos e por ser o pilar incontestável de minha família, ainda que inconscientemente.

A Deus, por me conceder o dom da vida, para que hoje eu possa estar agradecendo a todas estas pessoas.

A TODOS VOCÊS, MUITO OBRIGADO!

Alexandre Santos Brandão

*“No que diz respeito ao desempenho, ao compromisso, ao esforço, à dedicação,
não existe meio termo. Ou você faz uma coisa bem feita ou não a faz.”*

Ayrton Senna

Resumo

Este trabalho descreve um esquema de controle descentralizado para coordenar a navegação de uma formação líder-seguidor de robôs móveis, incluindo a possibilidade de evasão de obstáculos. É utilizado o sensoriamento por varredura laser para a realização de tarefas de posicionamento e deslocamento do robô seguidor em relação ao robô líder. Modificações no algoritmo de desvio tangencial (implementado no robô líder) são apresentadas e experimentalmente provadas para a evasão de obstáculos em configurações côncavas. Para um ambiente semi-estruturado, é desenvolvida uma estratégia de localização do robô líder através das medidas provenientes do sensor laser a bordo do robô seguidor, visto que no controle descentralizado de formação, caso aqui tratado, não há compartilhamento de informações. Estratégias de formação rígida e semi-rígida são aqui abordadas. Primeiramente, é descrita a formação rígida, na qual a estrutura líder-seguidor não pode ser deformada durante toda a navegação. Nesta abordagem, o robô líder é responsável pela navegação segura do robô seguidor. Posteriormente, é apresentada a formação semi-rígida, na qual a distância líder-seguidor não pode sofrer alteração durante a tarefa de desvio de obstáculo, ao contrário do ângulo de formação. Neste caso, um controle baseado em força fictícia é apresentado para alterar este ângulo, gerando-se um novo ângulo de referência para o robô seguidor, para permitir as manobras de evasão. Para validação de ambas as abordagens de formação são mostrados resultados simulados e experimentais, os quais ilustram a eficiência dos métodos supracitados em tarefas de cooperação. A estabilidade dos controladores foi comprovada pela convergência assintótica das variáveis de formação aos valores desejados durante os experimentos realizados, em conformidade com a análise teórica apresentada no projeto dos controladores.

Abstract

This thesis describes a decentralized control scheme proposed for guiding the navigation of a leader-follower formation of mobile robots, including the possibility that the whole formation avoid obstacles. A laser scanner is here adopted to provide the information necessary to accomplish tasks like positioning and displacement of the follower robot with respect to the leader one. Modifications in the tangential escape algorithm (implemented on the leader robot) are presented and experimentally tested for avoiding concave obstacles configurations. For semi-structured environments, a leader searching strategy is developed using the range measurements provided by the laser scanner mounted on the follower robot. In this leader-follower formation, there is no information sharing, emphasizing the decentralized control approach. Rigid and semi-rigid formation approaches are presented here. Firstly, a rigid formation is considered, which means that the leader-follower structure can not be deformed even during obstacle avoidance. In this case, the leader robot is responsible for the safe navigation of the follower robot. In the sequel, a semi-rigid formation is discussed, where the desired leader-follower distance can not be changed during obstacle avoidance, but the desired formation angle can be changed to make the maneuvers possible or even easier. Then, a control strategy based on a fictitious force is presented and implemented on the follower robot to allow changing the formation angle accordingly. To validate the proposed controller, regarding both formation approaches, simulation and experimental results are presented. Applying the described methods in the accomplishment of cooperation tasks, a very good system performance has been observed. Moreover, system stability can be verified through the asymptotic convergence of the formation variables to the desired values during the experiments, which is in accordance with the theoretical analysis performed when designing the controllers.

Sumário

Lista de Figuras

1	Introdução	14
1.1	Robôs Móveis a Rodas	18
1.1.1	Estrutura Omnidirecional	18
1.1.2	Estrutura <i>Car-Like</i>	19
1.1.3	Estrutura Uniciclo	19
1.2	Sistema Sensorial	20
1.3	Controladores de Posição e de Orientação	22
1.3.1	Controlador de Posição	23
1.3.2	Controlador de Orientação	25
1.3.3	Estabilidade de Sistemas Comutados	26
1.4	Objetivos deste Trabalho	27
1.5	Estrutura do Trabalho	28
2	Um Algoritmo para Desvio de Obstáculos: Desvio Tangencial	30
2.1	Desvio Tangencial	32
2.2	Desvio Tangencial Modificado	33
2.3	Resultados Experimentais e Discussão	40
2.4	Conclusões	47
3	Controle Descentralizado de Formação	48
3.1	Variáveis de Formação e Equações de Estado	51

3.2	Controle Descentralizado de Formação	53
3.2.1	Abordagem por Lyapunov	53
3.2.2	Abordagem por Cinemática Inversa	55
3.3	Estratégia Sensorial	57
3.3.1	Cálculo das Variáveis de Formação	57
3.3.2	Estratégia de Busca do Líder	61
3.4	Resultados e Discussão	63
3.4.1	Experimento 1 - Estacionamento	63
3.4.2	Experimento 2 - Busca de um alvo	64
3.4.3	Experimento 3 - Caminho circular	66
3.5	Conclusões	68
3.6	Trabalhos Futuros	69
3.6.1	Consideração sobre a Dinâmica do Robô Uniciclo Aplicada à Formação	69
3.6.2	Extensão para Formações com Múltiplos Robôs	72
3.6.2.1	Estrutura Líder-Seguidores	72
3.6.2.2	Estrutura Líderes-Seguidores	74
4	Desvio de Obstáculo Aplicado à Formação	79
4.1	Desvio de Obstáculos para uma Formação Rígida	81
4.1.1	Simulação e Discussão	81
4.1.2	Experimento e Discussão	83
4.2	Desvio de Obstáculos para uma Formação Semi-Rígida	86
4.2.1	Simulação e Discussão	89
4.2.2	Experimento e Discussão	91
4.3	Conclusões e Trabalhos Futuros	93
	Referências	95

Anexo A – Filtro de Kalman	99
A.1 Representação do Sistema	99
A.2 Otimização da Predição	102
Anexo B – Simulador Desenvolvido	104
B.1 Parte Gráfica	105
B.2 Sistema Sensorial	106
B.3 Estratégia de Desvio de Obstáculos	107
B.4 Estratégias Aplicadas ao Controle de Formação	107
B.5 Controladores	108
B.6 Exemplo	109
Anexo C – Plataforma Experimental - Pioneer 2DX	112

Lista de Figuras

1	Manipulador robótico industrial dotado de articulações rotacionais utilizado em atividades de soldagem.	14
2	Robôs móveis terrestres.	15
3	Diagrama de blocos da arquitetura de um robô móvel com raciocínio deliberativo.	16
4	Diagrama de blocos da arquitetura de um robô móvel com raciocínio reativo.	17
5	Robô móvel omnidirecional (<i>Palm-Pilot Robot Kit</i>).	18
6	Robô móvel a rodas tipo <i>car-like</i>	19
7	Robô móvel a rodas tipo unicycle (Pioneer 2DX da ActivMedia).	20
8	Sensor laser LMS 200 montado sobre o robô móvel Pioneer 2DX.	21
9	Medidas de distância tomadas com o sensor laser LMS 200.	22
10	Sistema de coordenadas para projeto do controlador de posição final.	24
11	Desvio Tangencial - Determinação do ângulo de rotação.	32
12	Diagrama de blocos do sistema de controle baseado na abordagem de desvio tangencial.	33
13	Modificação na abordagem de desvio tangencial para levar em conta obstáculos em forma de esquina.	34
14	Ajuste do ângulo de rotação γ	36
15	Ajuste do ângulo de rotação γ - representação geométrica.	37
16	Girando 90° após superar a extremidade do obstáculo.	37
17	Fluxograma do desvio tangencial modificado.	39
18	Fluxograma ilustrando uma segunda alternativa para tratar más orientações do robô móvel após vencer o obstáculo.	41

19	Desvio de um obstáculo em forma de U : (a,c,d) Giro em Malha Aberta (b,d,f) Giro com Controlador de Posição Final.	43
20	Desvio de um obstáculo em forma de L	44
21	Desvio de um obstáculo em forma de V	45
22	Desvio de um obstáculo em forma de Z com dois obstáculo circulares. . . .	46
23	Robôs móveis unicycle que compõem a formação linear.	50
24	Variáveis de formação.	51
25	Sensor de varredura laser utilizado na formação líder-seguidor.	58
26	O padrão a bordo do robô líder. (a) Ambiente de simulação da formação líder-seguidor. (b) Distância ρ_{LF} entre os robôs em função das medidas de laser.	58
27	Padrão montado sobre o robô líder. (a) Padrão virtual (padrão real como é visto pelo robô seguidor). (b) Definição do erro de orientação θ_{LF} e do ângulo de rotação aparente do padrão γ_2	59
28	Obtendo o ângulo β_{LF} através da odometria do robô seguidor.	60
29	Perfil das medidas do sensor laser no ambiente e a diferença entre medidas de distância consecutivas.	62
30	Experimento 1 - Estacionamento.	65
31	Experimento 2 - Controle de Posição em Formação.	67
32	Experimento 3 - Caminho circular por mudança de referência do ângulo de formação.	68
33	Diagrama de blocos do modelo dinâmico completo do robô móvel tipo uni- ciclo.	71
34	Estrutura de controle com compensação da dinâmica do robô móvel.	72
35	Formação Multi-robôs em losango.	73
36	Simulação: estrutura líder-seguidores com formação em losango.	75
37	Formação Multi-robôs em linha.	76
38	Simulação: estrutura líderes-seguidores em formação linear.	78

39	Abordagem de desvio tangencial aplicada à formação rígida.	81
40	Simulação 1 para a formação Rígida.	83
41	Experimento formação rígida com desvio de obstáculos.	85
42	Regiões de reação de cada robô em uma formação semi-rígida.	87
43	Estratégia de desvio de obstáculos baseado em impedância mecânica. . . .	88
44	Forças fictícias à direita e à esquerda do robô seguidor.	88
45	Simulação 1 para a formação Semi-rígida.	91
46	Simulação 2 para a formação Semi-rígida.	92
47	Experimento formação semi-rígida com desvio de obstáculos.	94

1 *Introdução*

“A beleza nas coisas existe apenas na mente que as contempla.”

(David Hume)

Tarefas cotidianas as mais diversas podem ser realizadas por robôs. Dentre elas podemos citar a limpeza doméstica e de galpões industriais, tarefas de montagem em linhas de produção, exploração de ambientes de risco e regiões inóspitas, aplicações militares de inspeção e de estratégia, tarefas agrícolas (agricultura de precisão), colaboração com humanos e cooperação entre robôs, por exemplo [1].

Em uma classificação geral de robôs, podemos especificar duas classes, a saber os robôs manipuladores (industriais) e os robôs móveis [2]. É importante salientar que apesar dos manipuladores robóticos apresentarem mobilidade durante a execução de uma tarefa esta mobilidade está limitada pelas dimensões físicas das articulações, definindo o que se conhece como espaço de trabalho do robô manipulador (Figura 1). Em contrapartida, os robôs móveis, dotados de seus meios específicos de locomoção, podem navegar em ambientes de grandes dimensões, com um certo grau de autonomia.



Figura 1: Manipulador robótico industrial dotado de articulações rotacionais utilizado em atividades de soldagem.

Os robôs móveis, por definição, são dispositivos mecânicos montados sobre uma base não fixa, capazes de interagir com o ambiente à sua volta através de seus sensores e atuadores, os quais são controlados por um sistema computacional normalmente a bordo [3]. Os robôs móveis são classificados quanto à sua anatomia, ou seja, em função do ambiente no qual eles foram projetados para navegar. Neste sentido, eles podem ser terrestres, aquáticos ou aéreos.

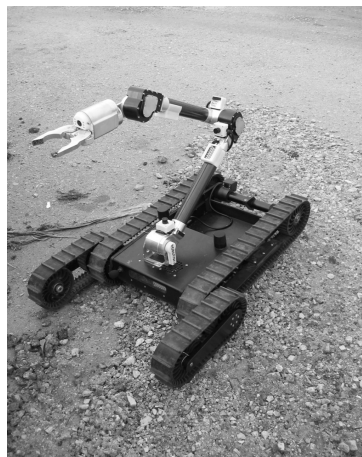
Dentre os mais difundidos na atualidade estão os robôs terrestres, os quais se deslocam através de movimentos de translação e rotação, e diferem entre si pelos diferentes tipos de atuadores utilizados para locomoção: rodas, esteiras ou patas. A Figura 2 ilustra robôs móveis terrestres e seus distintos meios de locomoção.

A escolha do tipo de locomoção depende do meio no qual o robô irá operar. Os robôs a rodas apresentam a vantagem de ser de fácil implementação e montagem, mas apresentam sérias restrições quanto a transposição de obstáculos, os quais devem ter diâmetros inferiores ao raio da roda. Robôs com esteiras são mais eficientes nas tarefas de desvio de obstáculos, porém dissipam muita energia devido ao atrito com o piso. Robôs com patas suprem as deficiências citadas, porém apresentam uma grande dificuldade no projeto dos controladores para coordenação dos movimentos das patas.

Nesta Dissertação são utilizados robôs móveis que se deslocam no ambiente através de rodas, com sistema de tração diferencial.



(a) Robô com rodas (projeto Robovlc da Universidade degli Studi di Catania, Itália, e Universidade de Leeds, Reino Unido)



(b) Robô com esteiras da indústria Robotic FX



(c) Robô com patas modelo TR35 da Universidade de Vermont.

Figura 2: Robôs móveis terrestres.

Quanto à sua estrutura, um robô móvel pode ser classificado como sendo de estrutura centralizada ou distribuída. Entende-se como sendo um robô de estrutura centralizada, aquele agente capaz de tomar suas próprias decisões com base nas informações coletadas a partir de seus próprios sensores. Este, porém, é incapaz de interagir com outros agentes presentes no mesmo espaço em que está inserido. Em contrapartida, um robô de estrutura distribuída refere-se a um agente que faz parte de uma sociedade e tem o seu papel bem definido na mesma. Seus atos e ações, portanto, não são tomados de forma isolada, e sim mediante a interação com os demais membros desta sociedade, compartilhando funcionalidades.

Quanto ao tipo de raciocínio de um robô móvel para sua interação com o ambiente, este pode ser classificado como deliberativo, reativo ou híbrido [4].

Entende-se por raciocínio deliberativo aquele no qual o robô móvel deve primeiramente obter as informações de seus sensores para reconstruir, o mais fielmente possível, o mundo em que irá navegar. Em seguida, baseado neste mundo, o algoritmo de controle inicia o planejamento das ações, a fim de realizar a tarefa que lhe foi estipulada. Por fim, os atuadores recebem os sinais de controle que, efetivamente, resultam no cumprimento da tarefa. A Figura 3 apresenta o digrama de blocos do raciocínio deliberativo. A eficiência deste tipo de raciocínio é degradada quando aplicado em ambientes dinâmicos, pois, neste caso, o planejamento se torna impraticável. Adicionalmente, é importante ressaltar que a reconstrução do mapa de navegação requer um elevado esforço computacional.

Por sua vez, entende-se por raciocínio reativo aquele que não requer informações prévias do mundo em que o robô está prestes a navegar. Não havendo, portanto, a etapa de reconstrução de um mundo virtual e, por conseguinte, a etapa de planejamento prévio das ações. A Figura 4 ilustra o diagrama de blocos deste tipo de raciocínio, o qual demonstra que cada comportamento (ou ação) é tratado de forma paralela. As informações correntes obtidas durante a navegação definem qual comportamento deverá ser excitado e, por fim,

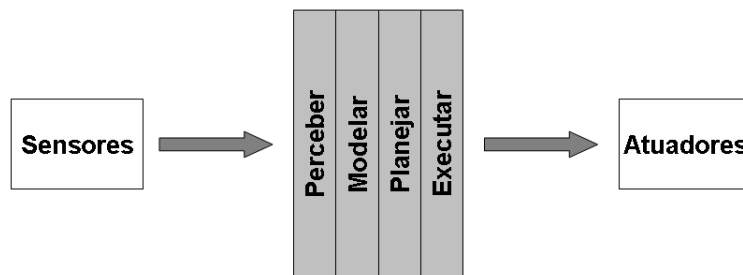


Figura 3: Diagrama de blocos da arquitetura de um robô móvel com raciocínio deliberativo.

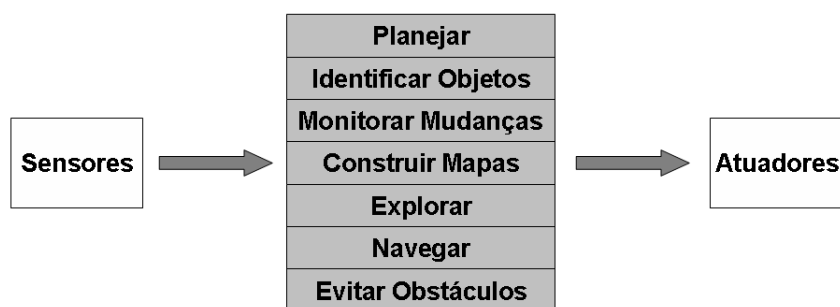


Figura 4: Diagrama de blocos da arquitetura de um robô móvel com raciocínio reativo.

executado. Este tipo de raciocínio permite ao robô navegar em ambientes cujos obstáculos podem se localizar em posições aleatórias. Afinal, tais obstáculos podem ser detectados e, de maneira reativa, evitados, quando necessário.

As dificuldades inerentes ao raciocínio reativo se devem basicamente ao paralelismo, pois torna-se inevitável a determinação da precedência de cada comportamento, tanto na execução das ações quanto na determinação de quais deles prevalecem durante o cumprimento de um objetivo.

Finalmente, entende-se por raciocínio híbrido aquele que engloba as funcionalidades das duas abordagens supracitadas, a fim de minimizar as deficiências apresentadas por cada uma, e, conseqüentemente, construir um sistema mais flexível e robusto.

A escolha do tipo de raciocínio a ser implementado em um robô móvel deve ser feita mediante a análise *a priori* do ambiente em que o mesmo irá navegar.

Por sua vez, o ambiente de navegação pode ser classificado como estruturado, semi-estruturado ou não estruturado, os quais se diferenciam entre si pela mobilidade dos objetos que o constituem.

Em um ambiente estruturado, a postura de qualquer objeto presente no espaço de trabalho do robô móvel é conhecida e não sofre alterações, ou seja, os objetos em cena são estáticos, sugerindo, portanto, uma abordagem deliberativa.

Em um ambiente semi-estruturado, admite-se um certo grau de mobilidade dos objetos presentes no meio. Escritórios, oficinas, laboratórios e residências são exemplos típicos de ambientes semi-estruturados, os quais sugerem a adoção de abordagem reativa.

Por fim, em aplicações nas quais não se tem controle sobre o deslocamento dos corpos, o robô móvel é dito estar inserido em um ambiente não estruturado, onde é inviável o planejamento de ações.

Nesta Dissertação, os robôs móveis são agentes centralizados, que navegam em ambientes semi-estruturados e possuem raciocínio implementado segundo a abordagem reativa.

1.1 Robôs Móveis a Rodas

1.1.1 Estrutura Omnidirecional

Construídos normalmente com uma separação angular entre suas rodas igual a 120° , os robôs móveis omnidirecionais podem executar movimentos em qualquer direção do plano **XY**. São freqüentemente dotados de rodas especiais, as quais são projetadas para propiciar o movimento do robô em direção perpendicular ao seu sentido de giro. O modelo cinemático deste tipo de robô é dado por

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{2}{3} \cos(\psi - \delta) & -\frac{2}{3} \cos(\psi - \delta) & \frac{2}{3} \sin \psi \\ \frac{2}{3} \sin(\psi - \delta) & -\frac{2}{3} \sin(\psi - \delta) & -\frac{2}{3} \cos \psi \\ \frac{1}{3L} & \frac{1}{3L} & \frac{1}{3L} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix},$$

onde $[x, y, \psi]^T$ representa a postura do mesmo no plano de navegação e o vetor \dot{q} representa a velocidade de cada uma das rodas. A distância entre as rodas e o ponto central do robô móvel é representada por L , e supõe-se que as rodas estão eqüidistantes a este ponto. O ângulo δ indica a orientação das rodas em relação ao sistema de coordenadas do robô. Vale mencionar que robôs móveis cuja configuração das rodas é omnidirecional não apresentam restrições não-holonômicas¹. A Figura 5 ilustra um robô omnidirecional.

¹Restrição não-holonômica é aquela que limita determinadas formas de caminhos (ou trajetórias) entre duas configurações distintas do ambiente, devido, basicamente, aos aspectos intrínsecos dos robôs utilizados. A título de informação, restrição holonômica é aquela que reduz o número de configurações possíveis que um robô pode atingir no ambiente.

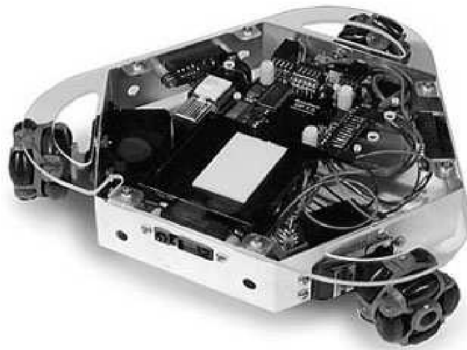


Figura 5: Robô móvel omnidirecional (*Palm-Pilot Robot Kit*).

1.1.2 Estrutura *Car-Like*

Semelhante a um automóvel convencional, a configuração *car-like* consiste em um par de rodas montadas sobre o mesmo eixo e tracionadas simultaneamente, as quais são localizadas, normalmente, na parte traseira do robô, e outro par de rodas (ou uma única roda) disposta na parte dianteira, que define a orientação (ou direcionamento) do robô durante a navegação. O modelo cinemático deste tipo de robô é dado por

$$\left\{ \begin{aligned} \dot{x} &= \nu \cos \psi, & \dot{y} &= \nu \sin \psi, & \dot{\psi} &= \nu \frac{\tan \xi}{L}, & \dot{\xi} &= \omega \end{aligned} \right\},$$

onde x e y definem a posição do robô no ambiente, visto que o ponto de interesse para posição é o ponto médio do eixo que une as rodas de tração. A orientação do robô é representada por ψ , ξ indica o ângulo de rotação das rodas dianteiras e L representa a distância entre as rodas dianteiras e traseiras. Os sinais de controle ν e ω são, respectivamente, as velocidades linear e angular do robô móvel. Este tipo de configuração apresenta restrições não-holonômicas que impossibilitam o movimento lateral, e, adicionalmente, para se orientar segundo um valor desejado de ψ , o robô necessita realizar movimentos de translação e rotação. Isto significa que o robô não pode girar sobre o próprio eixo. A Figura 6 apresenta um robô móvel tipo *car-like*.

1.1.3 Estrutura Uniciclo

Também chamados de robôs móveis de tração diferencial, estes robôs possuem as rodas montadas sobre um mesmo eixo (virtual), porém elas são acionadas independentemente. Este tipo de configuração apresenta a vantagem de permitir que o robô gire sobre o próprio eixo. O modelo cinemático do robô uniciclo é dado por

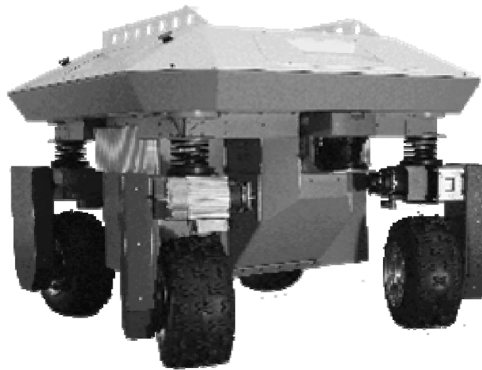


Figura 6: Robô móvel a rodas tipo *car-like*.

$$\left\{ \dot{x} = \nu \cos \psi, \quad \dot{y} = \nu \sin \psi, \quad \dot{\psi} = \omega \right\}, \quad (1.1)$$

onde x e y são as coordenadas do ponto localizado no centro do eixo que une as rodas do robô e ψ é a orientação do veículo em relação ao eixo x . As velocidades linear e angular do robô móvel, nomeadas ν e ω , respectivamente, representam os sinais de controle. Os robôs móveis tipo uniciclo apresentam a condição não-holonômica dada por

$$\dot{y} \cos \psi - \dot{x} \sin \psi = 0, \quad (1.2)$$

o que equivale a dizer que eles não podem se deslocar lateralmente. A Figura 7 ilustra um robô móvel com esta configuração.

Daqui em diante, quando um robô móvel for mencionado no texto, o leitor deve associá-lo àquela classe de robôs que se deslocam no ambiente utilizando rodas e são do tipo uniciclo de tração diferencial.

1.2 Sistema Sensorial

A autonomia durante a navegação de um robô móvel é ditada pela capacidade que este veículo possui de se deslocar no ambiente de forma segura, evitando colisões com possíveis obstáculos. Entretanto, para interagir com o ambiente, o robô necessita de um sistema sensorial capaz de fornecer medidas de proximidade entre os objetos presentes no meio e sua postura atual, sem a necessidade de contato. Dentre os sistemas mais difundidos para utilização em robótica móvel, podem-se destacar a visão computacional (câmaras de vídeo), sensores de ultra-som (sonares) e de varredura laser. O robô Pioneer



Figura 7: Robô móvel a rodas tipo uniciclo (Pioneer 2DX da ActivMedia).

2DX, ilustrado na Figura 7 e utilizado nos experimentos desta Dissertação, tem disponível todos estes dispositivos sensoriais.

Neste trabalho, o sensor de varredura laser foi eleito como sensor externo de proximidade. Segundo [5], o sensor laser é classificado como ativo, pois não depende de uma fonte de excitação externa para funcionar (caso oposto às câmaras CCD, por exemplo, que necessitam de excitação luminosa para fornecer as informações sobre o ambiente). Seu princípio de funcionamento consiste em emitir um feixe infravermelho e aguardar seu retorno após choque com algum anteparo. A Figura 8 ilustra o sensor LMS 200, fabricado pela empresa SICK, que é aquele que está instalado a bordo do robô Pioneer 2DX.

LMS² é a sigla para Sistema de Medição Laser, cujas aplicações envolvem operações de monitoramento, medição de distâncias, detecção e estimação da postura de objetos fixos ou móveis no ambiente, dentre outras. Os scanners LMS fornecem um mapa polar bi-dimensional, que relaciona a medida de distância com o ângulo em que tal medida foi realizada. Particularmente, a varredura em forma de leque mostrada na Figura 9 é construída pelo sensor LMS 200 através da reflexão do feixe de laser em um espelho rotativo. Os valores de distância podem ser tomadas em forma polar, em intervalos de 1°, 0,5° e 0,25°, cuja escolha é realizada via software. A Tabela 1 apresenta alguns dados característicos do sensor LMS 200.

²LMS, do inglês, *Laser Measurement System*.



Figura 8: Sensor laser LMS 200 montado sobre o robô móvel Pioneer 2DX.

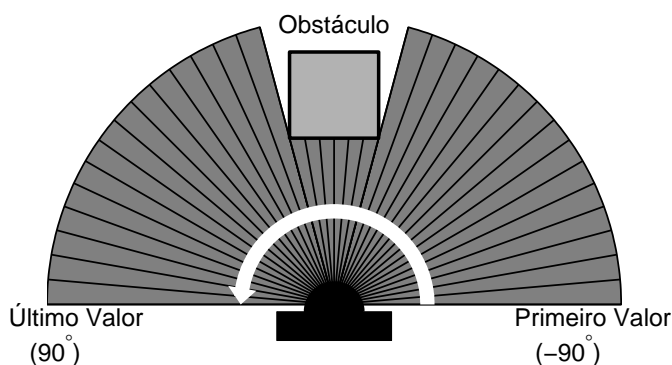


Figura 9: Medidas de distância tomadas com o sensor laser LMS 200.

Tabela 1: Características do sensor LMS 200

Aspectos Gerais	
Alcance Máximo	80m
Resolução Angular	0,25° / 0,5° / 1°, passível de seleção via software
Tempo de Resposta	53ms / 26ms / 13ms, relacionado à resolução angular
Erro estatístico	Tipicamente, 5mm para alcance $\leq 8m$, reflectividade $\geq 10\%$ e iluminação $\leq 5kLux$
Método de Cálculo da Distância	Tempo de Vôo (TOF, do inglês <i>Time of Flight</i>). $TOF = 2D/c$, onde c é a velocidade da luz no meio em que se propaga e D é a distância entre o sensor e a superfície refletora.
Aspectos Elétricos	
Interface para transferência de dados	RS 232/RS 422, configurável
Taxa de transferência de dados	9,6/19,2/38,4/500 kbaud
Tensão de Alimentação do Scanner e dos componentes eletrônicos	24 Vcc $\pm 15\%$ (ripple máximo de 500 mV), corrente máxima requerida de 1,8 A (incluindo as cargas inseridas)
Potência requerida	Aproximadamente 20W sem cargas de saída
Temperatura ambiente de operação	0 a +50 °C
Aspectos Mecânicos	
Peso	Aproximadamente 4,5 kg (desconsiderando o peso do suporte)

1.3 Controladores de Posição e de Orientação

Inicialmente, esta Dissertação tem o objetivo de propiciar a um robô móvel tipo unicycle uma navegação segura em ambientes semi-estruturados, utilizando a abordagem de controle reativo e um sensor de varredura laser. A proposta, portanto, é fazer com que o robô navegue entre dois pontos no ambiente evitando obstáculos. A navegação, por sua vez, deve ser realizada de forma estável, isto é, o robô deve sair das coordenadas de

origem e atingir assintoticamente o alvo desejado, e nele permanecer sem oscilações. A um controlador projetado segundo este objetivo dá-se o nome de *Controlador de Posição*. Entretanto, em algumas aplicações é desejável que o robô atinja o alvo com uma orientação pré-definida. A estes controladores dá-se o nome de *Controlador de Postura*, ou ainda Controlador de Posição e Orientação Final.

Esta seção apresenta um controlador de postura a ser implementado, nos capítulos subsequentes desta dissertação, no robô líder da formação. Duas abordagens podem ser tratadas no que toca ao controle de postura. A primeira delas entende-se por controle de posição e orientação final, onde o robô móvel, durante a navegação, atinge a posição final com a orientação final desejada, isto é, os sinais de controle enviados ao robô resultam em um caminho assintótico que o leva à postura desejada. Durante o projeto deste controlador são levadas em consideração todas as variáveis de estado que definem a postura do robô no ambiente, ou seja, $[x \ y \ \psi]^T \rightarrow [x_d \ y_d \ \psi_d]^T$ para $t \rightarrow \infty$. A segunda delas, entende-se por controle de posição e de orientação, cuja diferença se encontra no fato do robô atingir a posição desejada e, posteriormente, orientar-se para obter a postura desejada. Nesta abordagem, são projetados dois controladores distintos, um de posição final, que leva $[x \ y]^T \rightarrow [x_d \ y_d]^T$ para $t \rightarrow t_c$, e outro de orientação final, que faz $\psi \rightarrow \psi_d$ para $t_c \rightarrow \infty$, os quais comutam entre si para cumprir a tarefa proposta. O instante de tempo t_c representa o momento do comutação dos controladores.

Este trabalho adotou a abordagem de controle de posição e de orientação, a qual será descrita em detalhes nas subseções a seguir.

1.3.1 Controlador de Posição

O controlador de posição descrito nesta seção é apresentado em [6]. O intuito de comentar sobre este controlador se deve à implementação do mesmo no robô líder da formação linear tratada nos Capítulos 3 e 4.

A Figura 10 ilustra os referenciais que irão descrever o controle de posição. A origem do plano absoluto, ou referencial inercial, é representado por $\langle o \rangle$. O sistema de coordenadas do robô móvel é dado por $\langle a \rangle$ e o alvo desejado se encontra referenciado por $\langle g \rangle$. Assim, o objetivo do controle de posição é levar o robô móvel, $\langle a \rangle$, inicialmente localizado em $\langle o \rangle$, até $\langle g \rangle$, de forma estável.

Visualizando a Figura 10, os erros de posição podem ser definidos como

$$\{\tilde{x} = x_d - x, \quad \tilde{y} = y_d - y\}, \quad (1.3)$$

onde x_d e y_d representam as coordenadas do ponto desejado. Logo, levar os erros de posição a zero equivale dizer atingir o alvo desejado.

Ainda visualizando a Figura 10, o erro de posição ρ , a orientação do alvo θ em função da posição corrente do robô e o erro de orientação α do robô em relação ao alvo $\langle g \rangle$ são dados por

$$\left\{ \rho = \sqrt{\tilde{x}^2 + \tilde{y}^2}, \quad \theta = \arctan \frac{\tilde{y}}{\tilde{x}}, \quad \alpha = \theta - \psi \right\}. \quad (1.4)$$

Utilizando a Equação 1.4, o modelo cinemático do robô móvel apresentado na Equação 1.1 pode agora ser escrito em coordenadas polares, como

$$\left\{ \dot{\rho} = -\nu \cos \alpha, \quad \dot{\alpha} = \nu \frac{\sin \alpha}{\rho} - \omega, \quad \dot{\theta} = \nu \frac{\sin \alpha}{\rho} \right\}. \quad (1.5)$$

Desta forma, o controle de posição consiste em levar os erros de posição e de orientação a zero, ou seja, levar $[\rho \ \alpha]^T \rightarrow [0 \ 0]^T$, e, conseqüentemente, obter que $[x \ y]^T \rightarrow [x_d \ y_d]^T$.

O controlador aqui apresentado é projetado com base na Teoria de Lyapunov. Assim, é proposta a função candidata de Lyapunov radialmente ilimitada, dada por

$$V(\rho, \alpha) = \frac{1}{2}\rho^2 + \frac{1}{2}\alpha^2 > 0. \quad (1.6)$$

Tomando sua derivada temporal, tem-se, após substituir a Equação 1.5, que

$$\dot{V}(\rho, \alpha) = -\rho\nu \cos \alpha + \alpha \left(\nu \frac{\sin \alpha}{\rho} - \omega \right). \quad (1.7)$$

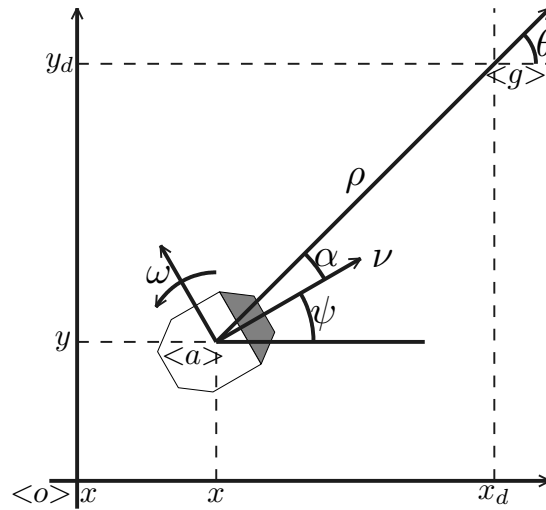


Figura 10: Sistema de coordenadas para projeto do controlador de posição final.

Adotando os sinais de controle

$$\begin{aligned}\nu &= \nu_{max} \tanh \rho \cos \alpha \\ \omega &= k_\omega \alpha + \nu_{max} \frac{\tanh \rho}{\rho} \sin \alpha \cos \alpha,\end{aligned}\tag{1.8}$$

onde ν_{max} e k_ω são constantes reais positivas. Para evitar a saturação dos controladores, os sinais de controle gerados são limitados, conforme estabelecido pela função de saturação tangente hiperbólico ($\tanh(\cdot)$).

Inserindo tais sinais de controle (Equação 1.8) na Equação 1.7, obtém-se

$$\dot{V}(\rho, \alpha) = -\nu_{max} \rho \tanh \rho \cos^2 \alpha - k_\omega \alpha^2 < 0,\tag{1.9}$$

ou seja, a derivada da função candidata de Lyapunov é definida negativa. Logo $[\rho \ \alpha]^T \in L_\infty^n \cap L_2^n$, e, portanto, garante-se que $[\rho \ \alpha]^T \rightarrow [0 \ 0]^T$ para $t \rightarrow \infty$, assintoticamente. É importante ressaltar que ρ deve ser ligeiramente diferente de zero, a fim de evitar uma singularidade na Equação 1.5. Assim, considera-se que o robô atingiu a posição desejada quando $\rho < \delta$, sendo $\delta > 0$ arbitrariamente pequeno, o qual irá sanar, portanto, uma possível situação de singularidade.

1.3.2 Controlador de Orientação

Em situações nas quais o robô móvel é dotado de uma garra, este extremo operativo deve estar orientado em relação ao objeto a ser agarrado. Neste contexto, é proposto a seguir um controlador de orientação, o qual deve rotacionar o robô até um ângulo desejado e mantê-lo nesta postura. Tal controlador é proposto em [7].

Pensando na situação em que o robô já atingiu o alvo desejado, utilizando, por exemplo, o controlador apresentado na Subseção 1.3.1, verifica-se que $\alpha \rightarrow 0$ para $t \rightarrow \infty$. Assumindo que o sistema se encontra em estado estacionário, tem-se que $\psi = \theta$ (ver Equação 1.4).

O objetivo do controlador de orientação é fazer com que $\psi \rightarrow \psi_d$. O erro de orientação é dado por

$$\tilde{\psi} = \psi_d - \psi.\tag{1.10}$$

Do modelo cinemático do robô móvel apresentado na Equação 1.1, tem-se que

$$\dot{\psi} = \omega.\tag{1.11}$$

Tomando a derivada temporal da Equação 1.10 e assumindo que ψ_d é uma constante, a Equação 1.11 é agora dada por

$$\dot{\psi} = -\omega. \quad (1.12)$$

Seja uma função candidata de Lyapunov que garanta a estabilidade assintótica do ponto de equilíbrio $\tilde{\varphi} = 0$ da Equação 1.12, ou seja,

$$V(\tilde{\psi}) = \frac{1}{2}\tilde{\psi}^2. \quad (1.13)$$

Tomando sua derivada e utilizando a Equação 1.12, resulta

$$\dot{V}(\tilde{\psi}) = -\tilde{\psi}\omega. \quad (1.14)$$

Assumindo os sinais de controle dados por

$$\begin{aligned} \nu &= 0 \\ \omega &= \omega_{max} \tanh \tilde{\psi}, \end{aligned} \quad (1.15)$$

para $\omega_{max} > 0$, obtém-se

$$\dot{V}(\tilde{\psi}) = -\omega_{max}\tilde{\psi} \tanh \tilde{\psi} < 0, \quad (1.16)$$

ou seja, a derivada da função candidata de Lyapunov é definida negativa e, por conseguinte, o equilíbrio $\tilde{\psi} = 0$ é assintoticamente estável. Assim, $\psi \rightarrow \psi_d$ para $t \rightarrow \infty$.

1.3.3 Estabilidade de Sistemas Comutados

Em um sistema de controle de postura utilizando os controladores apresentados nas Subseções 1.3.1 e 1.3.2, quando o robô atinge o ponto alvo o sistema deve comutar do controlador de posição para o de orientação. Porém, deve-se ter em mente que a estabilidade dos sistemas deve estar assegurada durante o chaveamento de controladores. Para isto, utiliza-se a extensão direta do teorema de estabilidade de Lyapunov, que garante a estabilidade de sistemas comutados desde que os controladores de cada sistema em separado sejam projetados utilizando a mesma função candidata de Lyapunov [8].

Quando o robô atinge a posição desejada, um valor de orientação desejada deve ser atingido. Pela Figura 10, isto equivale a dizer que $\theta = \psi_d$. Assim, o objetivo agora é fazer com que $\psi \rightarrow \theta$. Para verificar a estabilidade na comutação, considera-se o desenvolvimento do controlador de posição proposto na Seção 1.3.1 aplicado ao controle de orientação. Tomando os sinais de controle da Equação 1.8, porém assumindo $\rho \approx 0$,

tais sinais podem ser escritos como

$$\begin{aligned}\nu &= 0 \\ \omega &= k_\omega \alpha,\end{aligned}\tag{1.17}$$

ou seja, eles são similares à Equação 1.15. Pela estabilidade assintótica, já demonstrada na Seção 1.3.1, conclui-se, portanto, que $\alpha \in L_\infty^n \cap L_\infty^2$, ou seja, $\alpha \rightarrow 0$ para $t \rightarrow \infty$, e, por sua vez, $\psi \rightarrow \theta = \psi_d$, para $t \rightarrow \infty$, efetuando a re-orientação do robô para estabelecer a postura desejada. Adicionalmente, pode-se dizer que a função candidata de Lyapunov da Equação 1.13 está contida na da Equação 1.6, e, portanto afirmar que ambos os controladores foram projetados segundo a mesma função candidata de Lyapunov, o que garante a estabilidade de sistemas comutados.

1.4 Objetivos deste Trabalho

O presente trabalho visa um controle descentralizado de formação baseado na estrutura líder-seguidor. Ressalta-se que as informações sobre o estado da formação provêm unicamente do sensor de varredura laser a bordo do robô seguidor.

O objetivo global deste trabalho é fazer com que um robô móvel dotado de um sensor de varredura laser seja capaz de identificar e seguir um outro robô, mantendo uma distância e uma orientação relativa pré-definidas. Por fim, assegurar que ambos os robôs naveguem em formação e sejam capazes de evitar obstáculos durante o cumprimento da tarefa a eles atribuída. Para cumprimento de tal objetivo geral, estipulam-se alguns objetivos específicos adicionais, a saber:

- estudar o modelo cinemático do robô móvel tipo uniciclo e controladores de postura já projetados para estes robôs;
- propor uma estratégia de desvio de obstáculos capaz de superar obstáculos côncavos;
- desenvolver controladores para o controle descentralizado de formação segundo a estrutura líder-seguidor;
- propor uma técnica de identificação do robô líder (que tem sobre si um padrão) pelo robô seguidor (que tem a bordo um sensor de varredura laser), a fim de obter uma estimativa das variáveis de formação;
- agregar à formação a capacidade de desvio de obstáculos.

1.5 Estrutura do Trabalho

Esta Dissertação foi estruturada de forma a se ter capítulos auto-contidos, com suas próprias conclusões, e está organizada como segue:

Capítulo 1: Introdução

Este capítulo descreve, em linhas gerais, o tema desta dissertação. De forma não aprofundada, o capítulo comenta sobre a classificação de robôs móveis quanto à sua anatomia, estrutura de controle e tipo de raciocínio. Adicionalmente, apresenta o sistema sensorial utilizado para navegação e alguns controladores, já apresentados na literatura, os quais serão úteis, do ponto de vista de aplicação, nos capítulos subsequentes. Para encerrar o capítulo, são apresentados os problemas de controle a serem tratados nesta Dissertação, assim como os objetivos a serem cumpridos com este trabalho.

Capítulo 2: Um Algoritmo para Desvio de Obstáculos: Desvio Tangencial

Este capítulo discute, inicialmente, o algoritmo de desvio de obstáculos denominado *Desvio Tangencial*. Em seguida, são agregadas funcionalidades a esta estratégia, que possibilitam ao robô móvel superar obstáculos antes intransponíveis. O algoritmo passa a ser chamado de *Desvio Tangencial Modificado*. Para validação da técnica são apresentados resultados experimentais para diversas configurações de obstáculos, inclusive obstáculos côncavos. Por fim, são apresentadas algumas discussões e conclusões a respeito da estratégia de evasão de obstáculos.

Capítulo 3: Controle Descentralizado de Formação

Neste capítulo é introduzido o problema de controle descentralizado de formação, onde, inicialmente, são apresentadas as variáveis de formação e as equações de estado que regem o sistema líder-seguidor tratado nesta Dissertação. Em seguida, controladores projetados segundo a abordagem por Lyapunov e por Cinemática Inversa são propostas para realizar o controle descentralizado de formação. Posteriormente, é explicitada a estratégia sensorial utilizada para viabilizar o cumprimento da tarefa de formação linear. Para validação da proposta, são apresentados resultados experimentais da estratégia de controle descentralizado de formação abordada neste capítulo, assim como as discussões e conclusões pertinentes. Por fim, são apresentadas algumas propostas para trabalhos futuros.

Capítulo 4: Desvio de Obstáculos Aplicado à Formação

Este capítulo agrega ao controle descentralizado de formação a tarefa de desvio de obstáculos. São aqui apresentadas estratégias de evasão de obstáculos aplicadas a uma formação linear líder-seguidor configurada segundo uma estrutura rígida e semi-rígida. Resultados de simulação e experimentais são apresentados, a fim de demonstrar a eficiência de ambas as estratégias. Conclusões e trabalhos futuros compõem a parte final deste capítulo.

2 *Um Algoritmo para Desvio de Obstáculos: Desvio Tangencial*

*“No meio do caminho tinha uma pedra
Tinha uma pedra no meio do caminho...”*

(Carlos Drummond de Andrade)

Durante a navegação de um robô móvel à procura de um ponto alvo em um ambiente semi-estruturado, há que se considerar a hipótese de existência de obstáculos em sua rota de busca. Portanto, é essencial abordar estratégias de evasão capazes de evitar possíveis colisões e, assim, assegurar que o robô alcance o objetivo desejado.

Na literatura, várias abordagens têm sido desenvolvidas visando o cumprimento do propósito supracitado. Dentre elas podem ser citadas algumas clássicas, tais como Detecção de Bordas [9], Grade de Certeza [10], Campos Potenciais [11], Campo de Força Virtual (VFF) [12] e Histograma de Campo Vetorial (VFH) [13]; ou outras mais recentes, tais como Diagrama de Proximidade (ND) [14, 15], Método de Deformação Reativa de Trajetória (RPD) [16], Método Baseado na Velocidade de Curvatura (CVM) [17], Diagrama Polar (PD) [18], Vetor de Repulsão (VR) [19], Programação Dinâmica (DP) [20], Abordagens Baseadas em Redes Neurais [21], Paradigma de Guiagem (SP) [22], Sistemas de Decisão Fuzzy (FDMS) [23] e Abordagens de Controle Neuro-Fuzzy (NFC) [24].

Algumas destas abordagens solucionam o problema de evasão de obstáculos através de um planejamento *a priori* do movimento, desde que se tenha o conhecimento global do ambiente de trabalho do robô móvel. Este tipo de estratégia é baseada no paradigma deliberativo, mencionado no Capítulo 1, e pode ser exemplificada pelas técnicas VFH, VFF, CVM, RPD, SP, Grade de Certeza e Programação Dinâmica. Contudo, uma vez planejada a trajetória, não se permite alterações nas posições dos objetos no ambiente, ou seja, o ambiente deve ser fortemente estruturado. Esta característica ressalta a debilidade

deste paradigma, quando se depara com situações onde um obstáculo inesperado (não modelado) corrompe a trajetória inicialmente planejada do robô. Não existe, portanto, a garantia de evasão. Entretanto, para tratar tais situações, algumas abordagens, como as apresentadas em [15–18, 22], modificam temporariamente a trajetória através de uma reação à presença de um obstáculo não modelado. Este tipo de estratégia se entende como abordagem híbrida, a qual incorpora mutuamente paradigmas deliberativo e reativo.

Em outra vertente, encontram-se os algoritmos de evasão de obstáculos baseados estritamente no paradigma reativo, onde se assume que o robô móvel não tem nenhum conhecimento *a priori* do ambiente em que é posto para navegar. No entanto, ele reage à situação atual do espaço de trabalho, em função das medidas provenientes de seus sensores externos. Ao contrário do paradigma deliberativo, que necessita armazenar o mapa do ambiente com o respectivo plano de movimento, o sistema de navegação reativo exige baixo esforço computacional, porém um sistema sensorial apropriado e confiável. Mudanças no estado do ambiente não interferem na navegação, pois o robô as percebe e, em seguida, reage a estas. Destaca-se, portanto, a eficiência da abordagem reativa para navegação em ambientes semi-estruturados. São exemplos de algoritmos estritamente reativos os métodos de Detecção de Bordas e de Campos Potenciais, bem como aqueles apresentados em [14], [19], [21], [23] e [24].

Inicialmente, este capítulo apresenta um algoritmo de evasão de obstáculos puramente reativo, inicialmente proposto em [25] e [5], onde se compara e se valida o desempenho da estratégia proposta mediante a utilização de sensores de ultra-som e de varredura laser. A este algoritmo dá-se o nome de *Desvio Tangencial*. Em seguida, são propostas modificações aplicadas a este algoritmo, a fim de tratar situações nas quais o robô móvel deve realizar manobras muito próximas ao obstáculo e situações em que ele se depara com obstáculos côncavos. Após agregar tais funcionalidades, o algoritmo em questão passa a ser chamado de *Desvio Tangencial Modificado*. Na sequência, são apresentados resultados experimentais e discussões a respeito do método modificado proposto. Para encerrar o capítulo, por fim, são ressaltadas algumas conclusões.

É importante mencionar que o algoritmo de Desvio Tangencial Modificado aqui tratado será implementado no robô líder da formação para realizar as tarefas de evasão de obstáculo mantendo uma estrutura linear. Por esta razão é que o referido algoritmo é discutido neste capítulo.

2.1 Desvio Tangencial

A estratégia de evasão aqui considerada denomina-se *Desvio Tangencial*, pois seu fundamento é evitar obstáculos executando trajetórias tangenciais a estes. Na existência de impedimentos que exijam uma mudança de rota do robô móvel em sua busca pelo ponto de destino (alvo real), é gerado um alvo virtual temporário, a fim de transpor tais obstruções, o qual será o novo objetivo visado. Uma vez que os sensores externos não detectem mais a presença de obstáculos nas proximidades, o alvo real é retomado como objetivo de controle, e o robô móvel reinicia sua busca. A Figura 11 ilustra a idéia da técnica de desvio tangencial.

O ângulo de rotação do alvo real para a criação do alvo virtual, γ , é calculado com base no conjunto de medidas provenientes do sensor de varredura laser a bordo do robô móvel. De posse destas medidas, o sistema verifica se há alguma medida de distância entre robô e o ambiente que seja inferior a uma distância segurança predefinida, ou seja, $d < d_{obs}$. Caso esta condição seja satisfeita, é tomado o ângulo β correspondente à medida mínima de distância ao obstáculo, tendo em vista a orientação do robô móvel. Dado que o sensor laser utilizado apresenta uma resolução angular de 1° (um grau) no intervalo de $[0^\circ \ 180^\circ]$, o ângulo β pode ser mapeado no intervalo $[-90^\circ \ 90^\circ]$, onde $\beta < 0$ indica a presença de obstáculos à direita do robô e $\beta > 0$ indica a presença de obstáculos à sua esquerda. Uma vez obtido o valor de β , o ângulo de rotação do alvo virtual γ pode ser obtido por

$$\gamma = \begin{cases} \beta - \alpha - 90^\circ, & \text{se } \beta \geq 0 \\ \beta - \alpha + 90^\circ, & \text{se } \beta < 0, \end{cases} \quad (2.1)$$

onde α representa o erro de orientação do robô móvel, conforme mencionado na Subseção 1.3.1. Destaca-se que, quando $\alpha > 0$ tem-se o alvo real localizado à direita do eixo de movimento do robô móvel. Durante a realização das manobras de evasão, a orientação

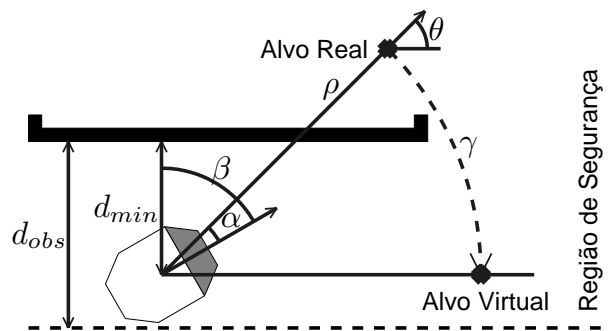


Figura 11: Desvio Tangencial - Determinação do ângulo de rotação.

do robô será paralela à tangente da fronteira do obstáculo, caracterizando, portanto, o desvio tangencial. Note-se que na Figura 11 o ângulo γ é negativo. Logo, a rotação do alvo real deve ocorrer em sentido anti-horário, de modo que o alvo virtual se localize à direita do robô móvel, considerando seu eixo de translação.

Lembrando que a navegação do robô móvel ocorre no plano bi-dimensional \mathbf{XY} , o ângulo γ é utilizado na matriz de rotação

$$\mathbf{X}_v = \begin{bmatrix} \cos \gamma & -\sin \gamma \\ \sin \gamma & \cos \gamma \end{bmatrix} \mathbf{X}_r \quad (2.2)$$

onde \mathbf{X}_r e \mathbf{X}_v definem a posição do alvo real (desejado) e a posição do alvo virtual, respectivamente. Logo, a Equação 2.2 ilustra a rotação do alvo real para sua nova posição \mathbf{X}_v , dando origem ao alvo virtual, conforme mostrado na Figura 11. Na ausência de obstáculos, não há modificação na posição do alvo real. Isto equivale a dizer que $\gamma = 0^\circ$, e o robô móvel prossegue em sua busca pelo alvo real.

O sistema de controle implementando o algoritmo de desvio tangencial é apresentado na Figura 12, onde ν , ω , ψ , \mathbf{X}_d e \mathbf{X} equivalem às mesmas variáveis inicialmente definidas na Seção 1.3, ao passo que d representa o vetor de medidas fornecidas pelo sensor de varredura laser. Tendo em vista o diagrama de blocos apresentado e a análise da estabilidade assintótica do controlador de posição, conclui-se que após se esquivar de todos os obstáculos o robô móvel sempre atingirá o alvo desejado, desde que este seja alcançável.

2.2 Desvio Tangencial Modificado

Em situações experimentais ou de simulação do algoritmo de desvio tangencial, existem algumas configurações de obstáculo, nas quais o robô móvel necessita realizar manobras complexas para evitar o obstáculo. Portanto, em alguns casos o choque é inevitável. Tais situações são tipicamente abordadas quando o robô deve superar obstáculos em es-

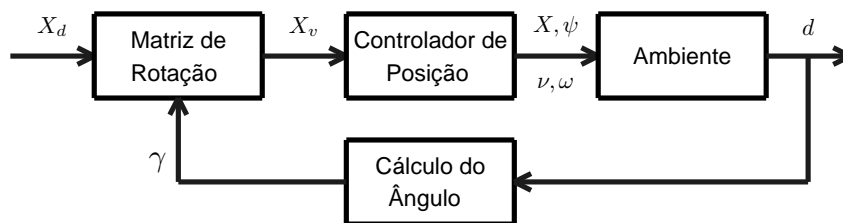


Figura 12: Diagrama de blocos do sistema de controle baseado na abordagem de desvio tangencial.

quina, como aqueles em forma de **L**, **V** e **U**. A Figura 13 ilustra um caso no qual o robô deve alcançar o alvo localizado abaixo da parte horizontal do obstáculo em forma de **L**.

Para evitar possíveis colisões, ou mesmo minimizar sua ocorrência, duas modificações foram introduzidas no algoritmo apresentado na Seção 2.1, daqui em diante denominado *Desvio Tangencial Convencional*. Tais modificações visam, primeiramente, evitar uma aproximação excessiva ao obstáculo, tendo em vista que a reação a este só se inicia dentro da região de segurança, e, em seguida, explorar a característica construtiva do robô móvel, isto é, o fato dele poder rotacionar sobre seu próprio eixo (robô tipo unicycle).

Na situação apresentada na Figura 13, o valor do ângulo de rotação do alvo virtual γ obtido pela Equação 2.1 não garante a evasão segura do obstáculo. Afinal, após rotacionar, o robô estará voltado para a parte vertical do obstáculo. Logo, ele deverá continuar girando. Isto quer dizer que o robô irá seguir de forma tangencial a parede horizontal, até que a distância à parede vertical seja a mínima. Para tratar esta situação, faz-se uma alteração no cálculo do ângulo γ , que agora será dado por

$$\gamma = \begin{cases} \beta - \alpha - 180^\circ & \text{se } \beta \geq 0 \\ \beta - \alpha + 180^\circ & \text{se } \beta < 0, \end{cases} \quad (2.3)$$

sempre que as medidas de distâncias d_{90° e d_{min} forem menores que d_{obs} . Entretanto, caso d_{90° seja maior que d_{obs} , γ é calculado segundo Equação 2.1. É importante salientar que d_{90° , mostrada na Figura 13, corresponde à medida laser referente à posição angular $\beta - 90^\circ$, se $\beta \geq 0$, ou $\beta + 90^\circ$, caso $\beta < 0$. A razão para utilizar ambas as medidas é dar ao robô móvel maior velocidade angular ω nas proximidades de obstáculos em forma de esquina.

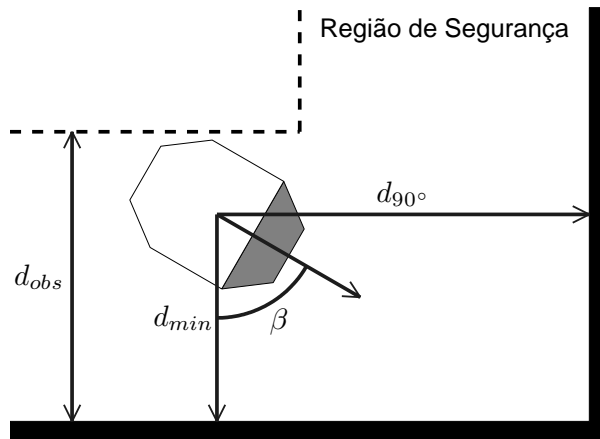


Figura 13: Modificação na abordagem de desvio tangencial para levar em conta obstáculos em forma de esquina.

Embora o robô móvel execute uma manobra de giro intensa, decorrente da modificação supracitada, a evasão segura ainda não está garantida. Afinal o robô deve executá-la com uma baixa velocidade linear ν , para maior segurança. Portanto, é necessário modificar a forma de cálculo da posição do alvo virtual, anteriormente expressa pela Equação 2.2, pois, conforme a Subseção 1.3.1, ν é diretamente proporcional ao erro de distância ρ , o qual é calculado por

$$\rho = \|\mathbf{X}_v - \mathbf{X}\|,$$

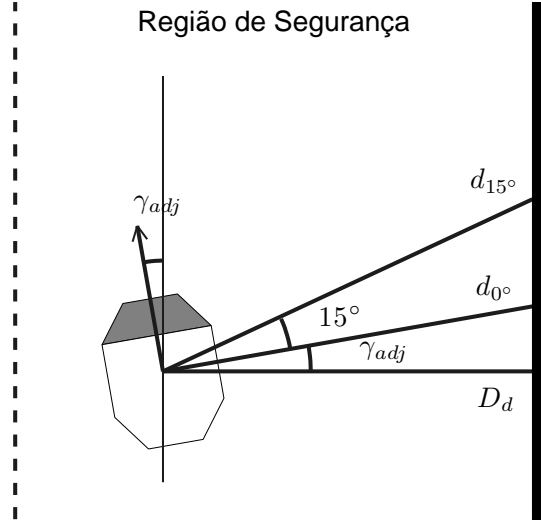
durante a evasão de obstáculos. Logo, utiliza-se

$$\mathbf{X}_v = \begin{bmatrix} \cos \gamma & -\sin \gamma \\ \sin \gamma & \cos \gamma \end{bmatrix} \begin{bmatrix} d_{min} \cos \theta \\ d_{min} \sin \theta \end{bmatrix} \quad (2.4)$$

para determinar a nova posição do alvo virtual, agora mais próximo ao robô móvel, com sua direção mantida (relembrando que θ representa a orientação do robô em relação ao alvo).

Por fim, após agregar tais modificações ao algoritmo de desvio tangencial convencional, as manobras de evasão passam a ser realizadas com menor velocidade linear e maior velocidade angular, reduzindo, pois, o risco de colisão. Tais feitos caracterizam a base do *Desvio Tangencial Modificado*.

Embora as modificações apresentadas sanem as debilidades da versão anteriormente proposta do algoritmo de desvio tangencial, algumas oscilações podem ocorrer durante o seguimento tangencial das bordas de um obstáculo. Isto ocorre devido a erros de medidas de distâncias inerentes ao sensor utilizado. Relembrando que o algoritmo de desvio tangencial modificado só é habilitado quando $d_{min} < d_{obs}$, durante o seguimento de uma parede, por exemplo, o robô pode sair da região de segurança e, por conseguinte, desativar o controlador de evasão de obstáculos prematuramente. Assim, para aumentar a robustez desta nova versão do escape tangencial, é proposta uma correção no ângulo de rotação do alvo, γ , em função da orientação do obstáculo perante o robô. A correção se assemelha à estratégia de seguimento de paredes apresentada em [5]. A Figura 14 apresenta uma parede localizada à direita do robô móvel, bem como feixes específicos de medidas de distância tomadas pelo sensor laser a bordo deste robô. O ângulo de orientação do robô em relação a parede é representado por γ_{adj} . A Figura 15 apresenta uma representação geométrica para cálculo de γ_{adj} .

Figura 14: Ajuste do ângulo de rotação γ .

Da Figura 15, tem-se que

$$\gamma_{adj} = \arctan \frac{a}{b} = \arctan \frac{d_{15^\circ} \cos 15^\circ - d_{0^\circ}}{d_{15^\circ} \sin 15^\circ},$$

onde d_{0° e d_{15° são as medidas de distância obtidas a 0° e 15° , respectivamente, da varredura do sensor laser. Vale ressaltar que para o cálculo de γ_{adj} o espaçamento entre os feixes laser foi assumido igual a 15° . Entretanto, poderia ser assumido qualquer valor inteiro menor que 90° , sendo que o limite de 90° se deve a que o mesmo procedimento deve ser realizado para o lado esquerdo do robô móvel. Portanto, a fim de contemplar a análise para obstáculos localizados em ambos lados do robô, tem-se que

$$\gamma_{adj} = \begin{cases} \arctan \frac{d_{15^\circ} \cos 15^\circ - d_{0^\circ}}{d_{15^\circ} \sin 15^\circ} & \text{se } \beta \geq 0 \\ -\arctan \frac{e_{15^\circ} \cos 15^\circ - e_{0^\circ}}{e_{15^\circ} \sin 15^\circ} & \text{se } \beta < 0, \end{cases} \quad (2.5)$$

onde e_{0° e e_{15° representam as medidas obtidas a 180° e 165° , respectivamente, da varredura do sensor laser, enquanto β indica a posição do obstáculo (à direita, se $\beta \geq 0$, ou à esquerda, caso contrário), conforme visto na Seção 2.1.

Por fim, a Equação 2.3 é alterada para

$$\gamma = \begin{cases} \beta - \alpha + \gamma_{adj} - 180^\circ & \text{se } \beta \geq 0 \\ \beta - \alpha + \gamma_{adj} + 180^\circ & \text{se } \beta < 0. \end{cases} \quad (2.6)$$

O ângulo de ajuste γ_{adj} é ignorado (ou assume-se zero) para obstruções não retilíneas que apareçam na rota de busca do robô móvel, como, por exemplo, obstáculos circulares.

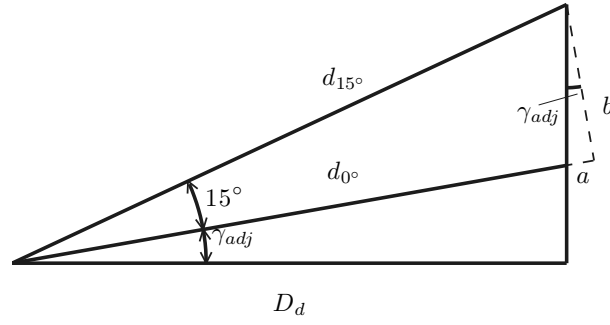


Figura 15: Ajuste do ângulo de rotação γ - representação geométrica.

Isto ocorre se a condição $d_{15^\circ} > \frac{d_{obs}}{\cos 15^\circ}$ for validada. Neste caso, o ângulo de rotação do alvo real γ volta a ser calculado pela Equação 2.3.

As alterações mencionadas até o presente momento garantem a evasão de obstáculos em forma de **L** ou **V**, dentre outras. Porém, para situações nas quais o robô móvel se encontra muito mal orientado, estas modificações são insuficientes para assegurar a transposição dos obstáculos. O obstáculo em forma de **U**, ilustrado na Figura 16, destaca esta situação, a qual ocorre no instante em que o robô finaliza o seguimento da parede lateral. Em tal situação seria assumida a transposição com sucesso do obstáculo pelo robô. O erro de orientação α do robô móvel após o seguimento é mostrado em linha contínua, destacando-se, assim, a má orientação do robô móvel em relação ao seu alvo.

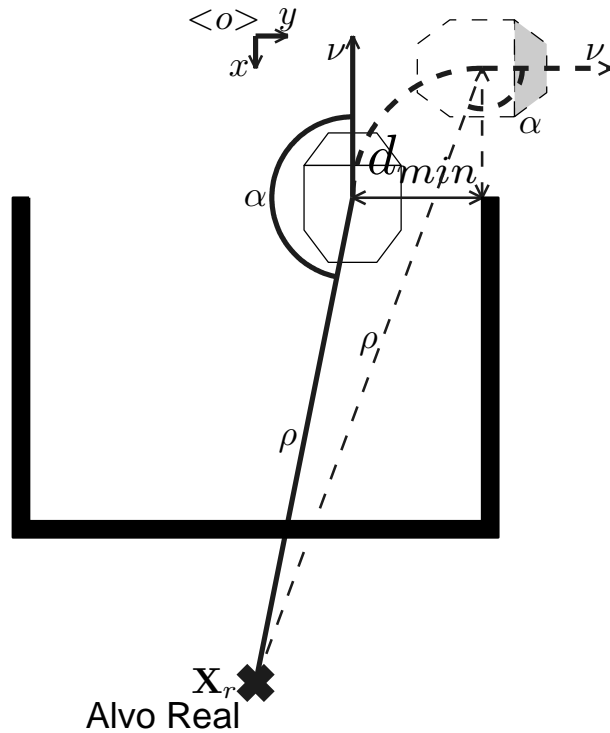


Figura 16: Girando 90° após superar a extremidade do obstáculo.

Para definir a presença (ou a ausência) de um obstáculo no ambiente, foi utilizado um *flag* chamado P_{obs} , o qual é ativado toda vez que alguma medida laser é inferior a d_{obs} , e desativado em caso contrário. Analisando simplesmente sua transição de estados, é possível determinar o instante em que o robô móvel encontrou um obstáculo ou o deixou para trás. Especificamente, para uma borda de descida subentende-se que o obstáculo foi vencido pelo robô. Este é o fundamento principal utilizado para evitar obstáculos côncavos.

Ao final de uma tarefa de evasão de obstáculo, ocorre uma borda de descida em P_{obs} . Neste instante, o valor de α é checado. Caso $|\alpha| > 90^\circ$, como de fato sucede na Figura 16, o robô deve percorrer a quarta parte de um arco de circunferência de raio d_{min} , girando, assim, para fora do obstáculo.

Duas abordagens são apresentadas nesta Dissertação para execução deste arco. A primeira delas visa manter constante a velocidade linear atual ν do robô, enquanto sua velocidade angular assume o valor $\omega = \frac{\nu}{d_{min}}$. Uma vez em movimento, o robô deve atingir a posição caracterizada em linha tracejada ilustrada na Figura 16. O intervalo de duração do movimento, no qual as ações de controle do robô são mantidas constantes, é calculado pela razão entre o perímetro do quarto de circunferência de raio d_{min} e a velocidade linear ν do robô, ou seja, $t_{travel} = \frac{0,5\pi d_{min}}{\nu}$. É importante mencionar que o último valor de β deve ser verificado, para indicar o sentido de giro. Caso $\beta < 0$, o robô deverá executar o giro em sentido horário, caso contrário, o giro será executado em sentido anti-horário. Ao final do giro, o robô se encontrará em uma situação mais favorável, pois apresentará um menor erro de orientação α . Posteriormente, o alvo real é retomado, bem como sua busca. Como resultado, o robô está apto a escapar de um problema de mínimo local característico de obstáculo côncavos (em forma de **U**, por exemplo). Note-se, porém, que esta abordagem se baseia em um controle em malha aberta.

A fim de contemplar por completo a abordagem de desvio tangencial modificado supracitada, a Figura 17 ilustra o fluxograma do algoritmo proposto. A parte esquerda deste fluxograma corresponde à manobra ilustrada na Figura 16.

A segunda abordagem visa levar o robô à posição caracterizada em linha tracejada na Figura 16 através da busca de um alvo temporário localizado nesta posição. Durante a busca por este novo alvo, o robô supervisiona o ambiente utilizando o próprio algoritmo de desvio tangencial modificado. Tal estratégia é ilustrada no fluxograma da Figura 18.

Após deixar o obstáculo para trás (borda de descida em P_{obs}), o novo alvo temporário $\begin{bmatrix} x_{alo} & y_{alo} \end{bmatrix}^T$ e a orientação do robô após alcançá-lo, ψ_{alo} , são definidos. Com base no

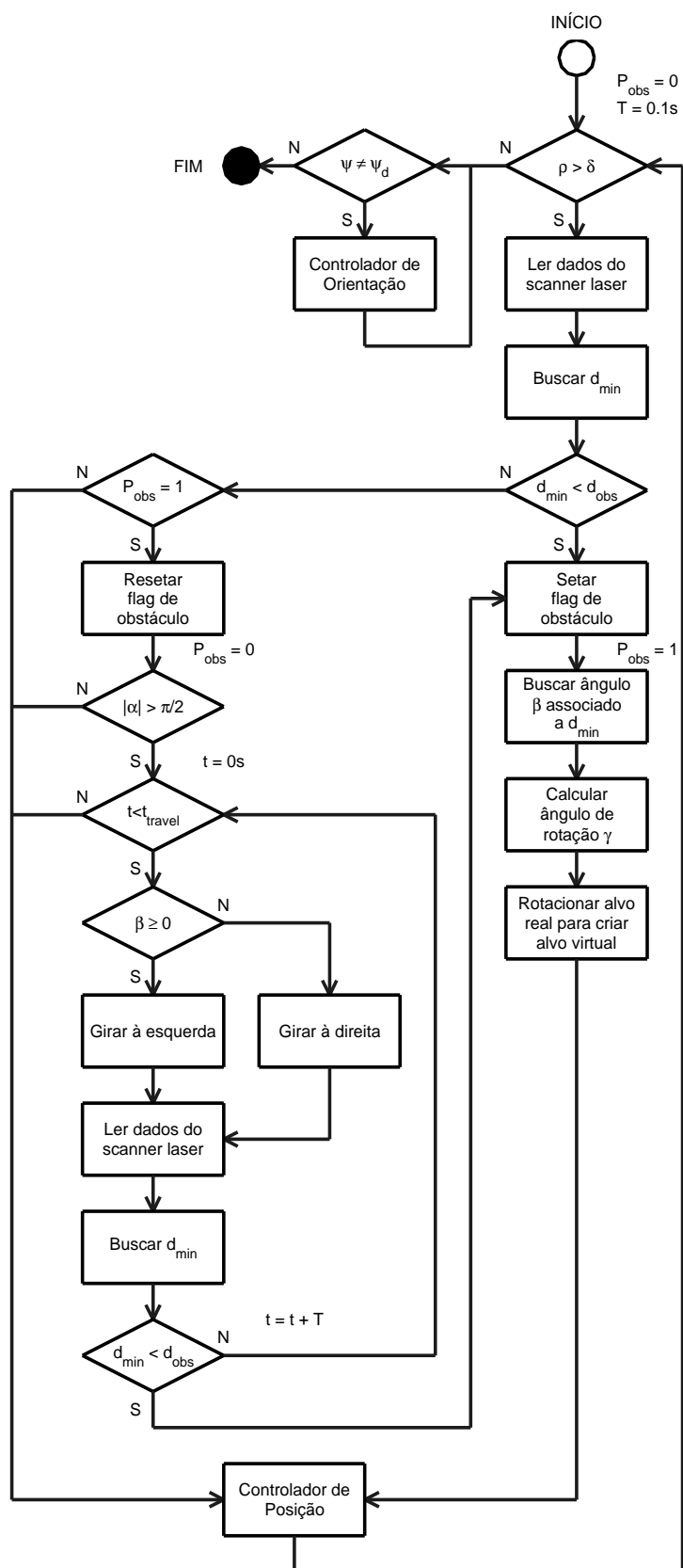


Figura 17: Fluxograma do desvio tangencial modificado.

valor de β , o qual define o sentido da execução do giro, a posição do novo alvo temporário é definida como sendo

$$\begin{bmatrix} x_{alo} \\ y_{alo} \end{bmatrix} = \begin{cases} \begin{bmatrix} x \\ y \end{bmatrix} + d_{min} \begin{bmatrix} -\sin\psi & \cos\psi \\ \cos\psi & \sin\psi \end{bmatrix} & \text{se } \beta \geq 0 \\ \begin{bmatrix} x \\ y \end{bmatrix} + d_{min} \begin{bmatrix} \sin\psi & \cos\psi \\ -\cos\psi & \sin\psi \end{bmatrix} & \text{se } \beta < 0, \end{cases} \quad (2.7)$$

onde $\begin{bmatrix} x & y \end{bmatrix}^T$ corresponde à posição atual do robô móvel e ψ é sua orientação, ambas com base no referencial inercial. Uma vez definido o alvo temporário, o robô inicia sua busca utilizando o algoritmo descrito na Subseção 1.3.1. Entretanto, caso algum obstáculo apareça durante a busca pelo alvo temporário, o processo é abortado e o alvo real é retomado.

Supondo que o robô atinja o novo alvo temporário sem qualquer impedimento, faz-se necessário uma re-orientação, a fim facilitar a fuga, bem como para posicioná-lo conforme a ilustração, em linha tracejada, da Figura 16. Tal correção é realizada através do Controlador de Orientação discutido na Subseção 1.3.2, cujo ângulo desejado é dado por

$$\psi_{alo} = \begin{cases} \psi + 90^\circ & \text{se } \beta \geq 0 \\ \psi - 90^\circ & \text{se } \beta < 0, \end{cases} \quad (2.8)$$

onde ψ é o valor de orientação do robô imediatamente antes de desabilitar P_{obs} . Após a etapa de re-orientação, o robô se encontra na posição desejada e com a orientação desejada, e, logo, o alvo real é reassumido e sua busca prossegue. Note-se, agora, que esta abordagem se baseia em um controlador secundário de posição e re-orientação.

2.3 Resultados Experimentais e Discussão

Visando corroborar a eficácia da abordagem de *Desvio Tangencial Modificado*, vários experimentos foram realizados. Entretanto, somente quatro deles são mencionados e discutidos neste trabalho de Dissertação. Os experimentos foram realizados com o robô móvel Pioneer 2DX da ActivMedia dotado de um sensor SICK de varredura laser. Foram assumidos os seguintes valores para as constantes envolvidas nos controladores de posição e orientação: $\nu_{max} = 0,35m/s$, $k_\omega = 0,40rad/s$ e $\omega_{max} = 0,45rad/s$. A distância de reação ao obstáculo (que delimita a região de segurança) foi assumida como $d_{obs} = 700mm$.

O primeiro experimento visa atingir um obstáculo localizado atrás de um obstáculo em

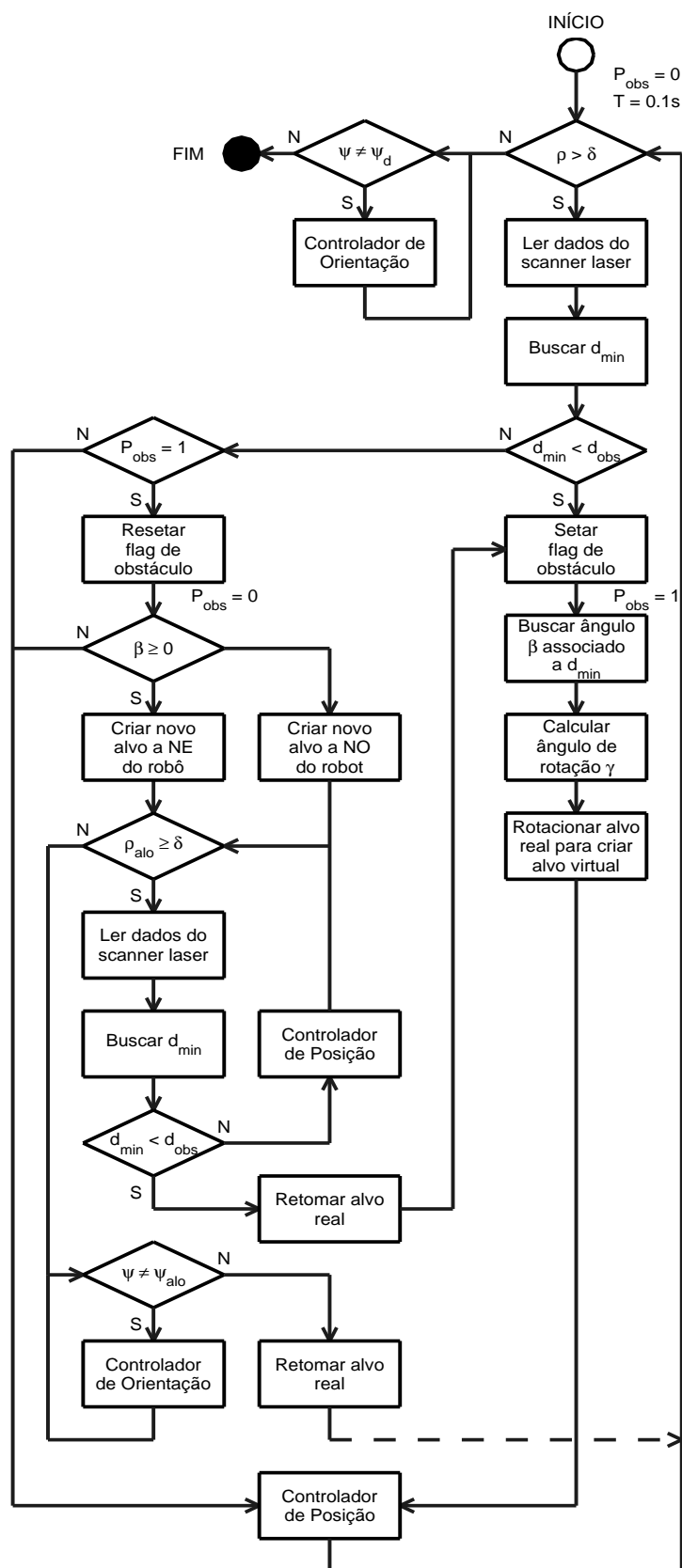


Figura 18: Fluxograma ilustrando uma segunda alternativa para tratar más orientações do robô móvel após vencer o obstáculo.

forma de **U**. Isto é, atingir, no plano **XY** um alvo na posição $(0mm, 3500mm)$ com uma orientação final igual a zero. Para esta configuração de obstáculo, as abordagens do algoritmo de desvio tangencial ilustradas nas Figuras 17 e 18 foram testadas e comparadas. A Figura 19(a) ilustra a trajetória percorrida pelo robô móvel, com o ambiente reconstruído a partir das medidas provenientes do sensor laser durante a execução da tarefa. As Figuras 19(c) e 19(e) ilustram o erro de posição, o erro de orientação e a orientação correntes do robô, e as ações de controle ν e ω , respectivamente, referentes ao experimento com a abordagem de controle em malha aberta. Por sua vez, as Figuras 19(b), 19(d) e 19(f) ilustram a trajetória percorrida e as mesmas variáveis, agora para o experimento usando a abordagem de controle secundário de posição e re-orientação. Constata-se que ambas as abordagens lograram executar por completo a tarefa de posicionamento e orientação. Entretanto, verifica-se um menor tempo de execução quando se utiliza o algoritmo apresentado na Figura 17, pois neste caso não é necessária a etapa de re-orientação (através do controlador de orientação de Subseção 1.3.2) após atingir o alvo temporário. Esta situação pode ser visualizada no intervalo de 32s a 40s das Figuras 19(d) e 19(f). Esta é a razão pela qual foi adotado o algoritmo da Figura 17 para os experimentos subsequentes.

Para completar a análise do experimento, em ambas as abordagens pode-se verificar que os erros de posição e de orientação tendem assintoticamente a zero, conforme proposto na análise de estabilidade descrita na Subseção 1.3.1. Por sua vez, a orientação do robô móvel tende ao valor desejado (zero) assintoticamente, segundo a proposta de controle de orientação descrita na Subseção 1.3.2. No tocante às ações de controle ν e ω , estas tendem assintoticamente a zero à medida que o robô se aproxima da posição desejada com a orientação estipulada, em decorrência da dependência direta dos erros das variáveis controladas (vide Equação 1.8).

Os resultados experimentais de uma navegação em um ambiente contendo um obstáculo em forma de **L** são mostrados na Figura 20. A tarefa a ser cumprida é alcançar um alvo localizado na posição $(3000mm, 3500mm)$, com uma orientação final igual a zero. A trajetória percorrida pelo robô e a reconstrução de seu espaço de trabalho são apresentadas na Figura 20(a). É possível verificar a trajetória cautelosa realizada para evitar o obstáculo em configuração de esquina, característica do *desvio tangencial modificado*. As ações de controle enviadas e efetivamente executadas pelo robô são ilustradas na Figura 20(c). Ao final da tarefa, estas ações tendem a zero em decorrência de sua dependência em relação às variáveis de controle (vide Figura 20(b)). Afinal, no sentido de Lyapunov, a estabilidade do controlador de posição e de orientação propostos garante a convergência assintótica aos valores desejados. Vale destacar que somente ajustes ligeiros foram realizados pelo

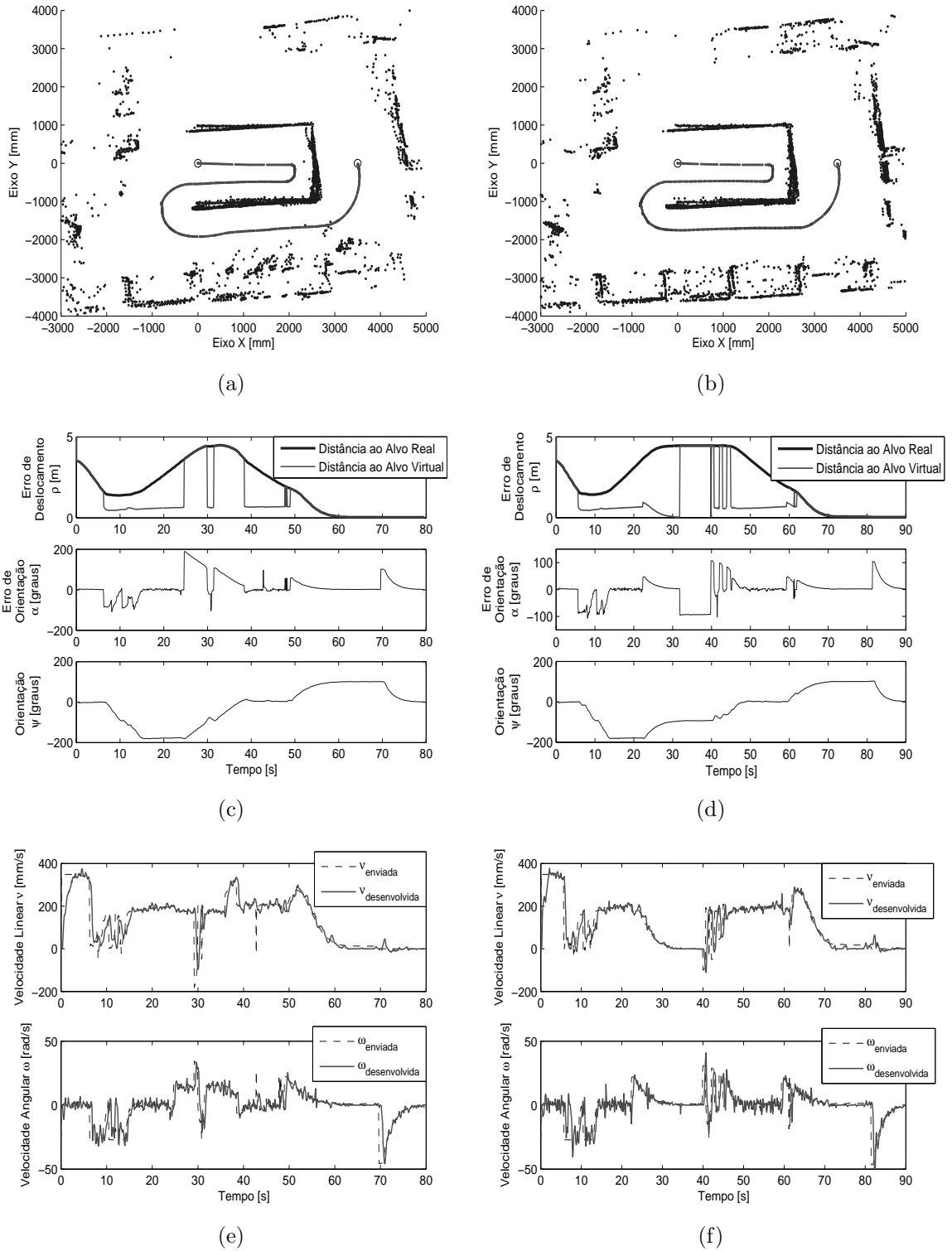
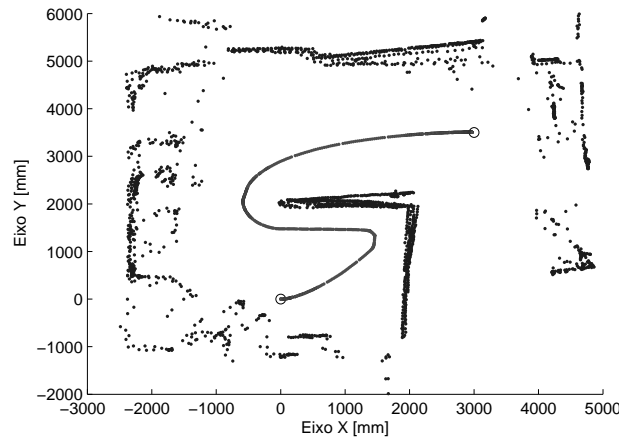


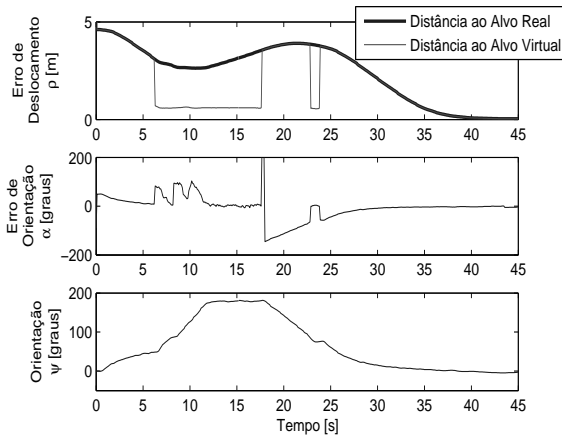
Figura 19: Desvio de um obstáculo em forma de **U**: (a,c,d) Giro em Malha Aberta (b,d,f) Giro com Controlador de Posição Final.

controlador de orientação (vide Figuras 20(b) e 20(c)), para que o valor desejado fosse atingido em decorrência da boa orientação de chegada do robô móvel ao atingir o alvo.

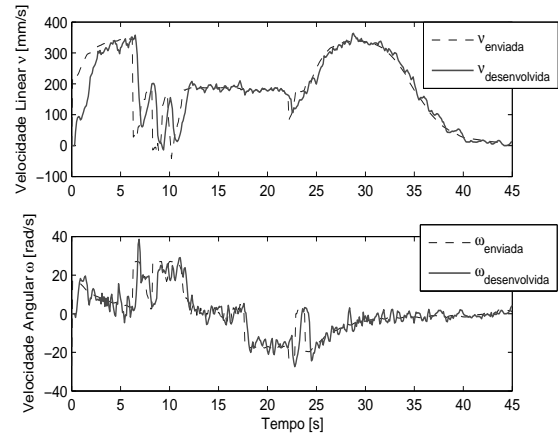
Um obstáculo em forma de **V** é a nova barreira a ser vencida pelo robô móvel, o qual



(a) Trajetória percorrida.



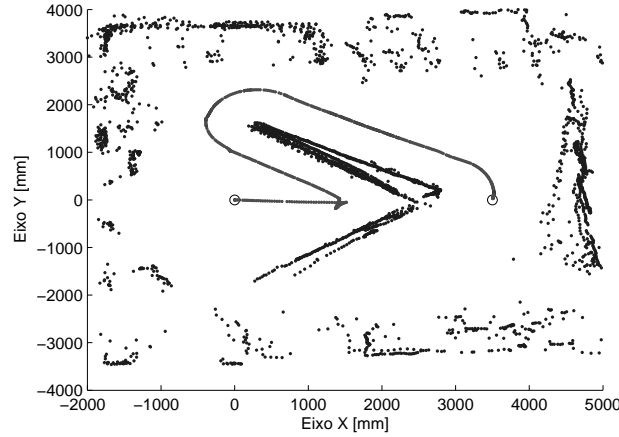
(b) Variáveis de controle.



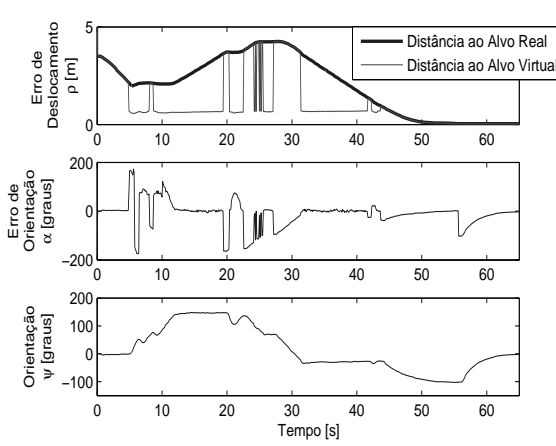
(c) Ações de controle.

Figura 20: Desvio de um obstáculo em forma de L.

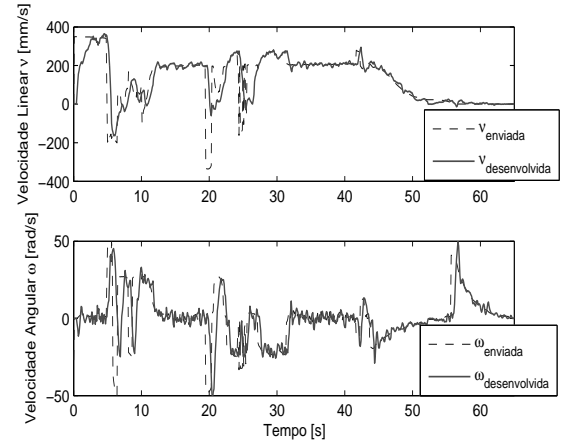
deve atingir a posição $(3500\text{mm}, 0\text{mm})$, com uma orientação igual a zero. Os resultados experimentais são ilustrados na Figura 21. A trajetória realizada pelo robô móvel é mostrada na Figura 21(a). Verifica-se a execução de manobras antes do seguimento da parede superior. Isto ocorre devido à forma do obstáculo: como o algoritmo implementado é puramente reativo, a detecção da distância mínima ocorre ora à direita ora à esquerda do robô móvel, levando o mesmo para o vértice do obstáculo. Entretanto, a modificação ilustrada na Figura 13 auxilia a decisão e fornece ao robô uma maior velocidade angular, essencial para fugir do obstáculo. A Figura 21(b) ressalta as variáveis de controle e sua convergência aos valores desejados, conforme estabelecido na análise de estabilidade. A Figura 21(c) ilustra as ações de controle envolvidas no cumprimento da tarefa proposta. Após 55s de experimento, verifica-se uma modificação abrupta na velocidade angular ω , a qual ocorre em função do chaveamento do controlador de posição para o de orientação, pois o orientação de chegada no ponto desejado ($\psi \approx 90^\circ$) é bem diferente do valor



(a) Trajetória percorrida.



(b) Variáveis de controle.

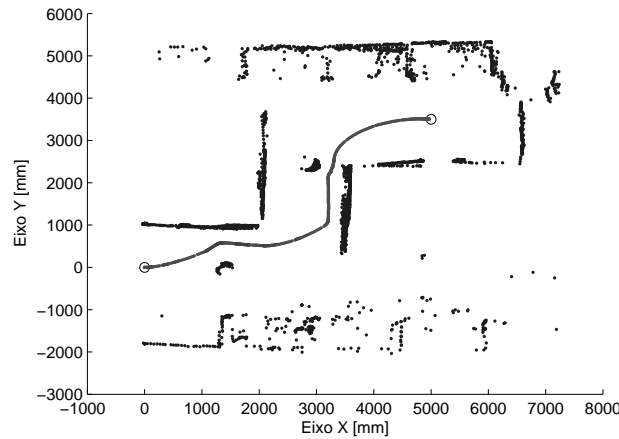


(c) Ações de controle.

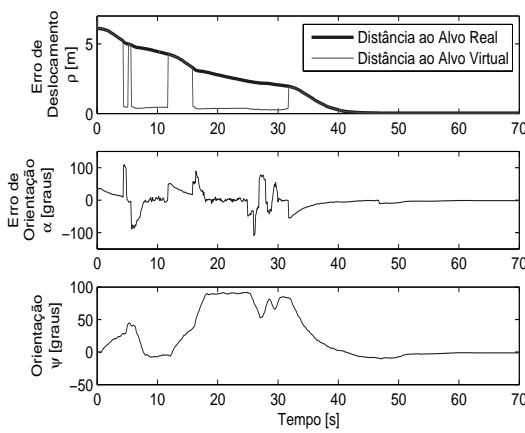
Figura 21: Desvio de um obstáculo em forma de **V**.

desejado ($\psi_d = 0$). Ao final da re-orientação, as ações de controle convergem novamente a zero e as variáveis de estado e a orientação do robô tendem aos valores desejados.

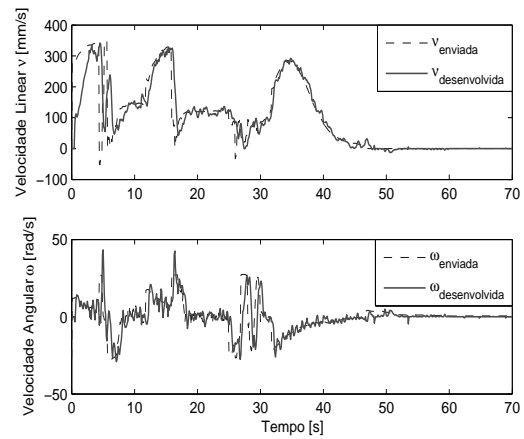
A Figura 22 apresenta os resultados de um experimento realizado em um corredor em forma de **Z** contendo dois obstáculos circulares. O alvo a ser atingido está localizado na posição $(5000mm, 3500mm)$, e o robô móvel deve atingi-lo com orientação final igual a zero. Os obstáculos de $300mm$ de diâmetro estão localizados nas posições $(1500mm, 0mm)$ e $(3000mm, 2500mm)$. Diferentemente dos experimentos anteriores, $d_{obs} = 500mm$, devido à distância entre os obstáculos e as paredes do corredor. A Figura 22(a) mostra a trajetória percorrida pelo robô móvel para atingir o objetivo desejado. É possível verificar que o robô obtém êxito, evitando os obstáculos em sua rota. As manobras são realizadas com cautela (baixa velocidade linear ν) nos pontos de estreitamento, conforme mostrado nos intervalos de $5 - 12s$ e $16 - 32s$ da Figura 22(c), a qual apresenta as ações de controle enviadas e efetivamente desenvolvidas pelo robô. Os erros de posição e orientação, bem



(a) Trajetória percorrida.



(b) Variáveis de controle.



(c) Ações de controle.

Figura 22: Desvio de um obstáculo em forma de **Z** com dois obstáculo circulares.

como a orientação do robô, são mostrados na Figura 22(b). A convergência assintótica de tais variáveis aos valores desejados pode ser visualizada, corroborando, assim, a análise de estabilidade via teoria de Lyapunov. Poucas variações ocorrem nas variáveis de controle e nos erros de posição e orientação após 50s. Isto se deve ao fato de que o robô móvel atingiu o alvo desejado com uma orientação próxima ao valor desejado, e por isto ações de controle muito sutis foram necessárias por parte do controlador de orientação para obter o valor desejado de orientação ψ_d .

O algoritmo de *desvio tangencial modificado* apresentado neste Capítulo é puramente reativo, simples e requer baixo esforço computacional. Para a tarefa de levar o robô móvel de um ponto inicial até um alvo de destino, evitando obstáculos que surgem em sua rota de busca, a técnica proposta se destaca perante outras apresentadas na literatura, as quais abordam problemas semelhantes. Em aspectos comparativos, um menor nível de extração de características é aqui requerido, quando comparado à abordagem proposta

em [19]. Com relação à técnica proposta em [14], a abordagem aqui descrita não necessita processar uma árvore de decisão para determinar a próxima ação, o que permite reagir de forma mais rápida à presença de obstáculos. Quanto ao ambiente de navegação, a proposta apresentada em [18] requer de antemão o conhecimento das dimensões dos obstáculos, a fim de modelá-los como cilindros a serem evitados, ao passo que na abordagem aqui proposta não se requer qualquer conhecimento *a priori* do ambiente semi-estruturado em que o robô é posto para navegar. No que diz respeito aos parâmetros ajustáveis, o algoritmo proposto envolve simplesmente d_{obs} , enquanto a técnica apresentada em [23] envolve vários parâmetros implicitamente acoplados. Por fim, o algoritmo aqui proposto não requer uma etapa de treinamento, como ocorre no sistema de controle neuro-fuzzy proposto em [24] para evasão de obstáculos.

Todo processamento lógico e matemático do algoritmo proposto ocorre em um intervalo de tempo inferior à leitura dos dados sensoriais, visto que o sensor de varredura laser utilizado para validação da técnica proposta apresenta uma frequência de amostragem de $10Hz$, ou seja, um vetor de 181 medidas é fornecido a cada intervalo de $0,1s$. Portanto, além de ser uma abordagem simples, o *desvio tangencial modificado* exige baixo esforço computacional, o que o torna uma proposta bastante atrativa quando comparada a outras abordagens similares. Adicionalmente, pode-se dizer que não haverá navegação carente de informações sensoriais, o que resulta em um deslocamento seguro do robô móvel em um ambiente parcialmente estruturado.

2.4 Conclusões

O controlador proposto na Seção 1.3 é estável no sentido de Lyapunov, de forma que todo alvo alcançável é alcançado pelo robô móvel. Logo, em situações de impedimento da rota de busca do alvo desejado, uma estratégia de evasão de obstáculo torna-se indispensável. Neste sentido, é apresentada neste Capítulo a abordagem de desvio tangencial, cuja essência é executar trajetórias tangenciais às bordas destes obstáculos.

Ao agregar as modificações propostas na Seção 2.2, o robô móvel logrou evitar as mesmas configurações de obstáculos antes evitadas pelo algoritmo de desvio tangencial descrito na Seção 2.1, e, adicionalmente, superar obstáculos antes intransponíveis.

Apesar de ser uma abordagem simples e de baixo esforço computacional, o algoritmo denominado *desvio tangencial modificado* é uma proposta eficiente e bastante atrativa em tarefas de controle de posição e orientação com evasão de obstáculos para ambientes semi-estruturados, quando comparada a abordagens similares apresentadas na literatura.

3 *Controle Descentralizado de Formação*

*“Nenhum organismo é uma ilha,
cada um se relaciona com outros organismos,
direta ou indiretamente.”*

*Simbiose
(Introdução às Relações Biológicas)*

Na literatura, a maioria dos pesquisadores que trabalham com robôs móveis estão conscientes que determinadas tarefas são melhor executadas por um grupo de robôs agindo de forma cooperativa ao invés de um único robô. Isto se deve à própria natureza da tarefa a ser cumprida ou ao custo associado, se a mesma for realizada por um robô especialista. Exemplos de tarefas mais compatíveis para a execução por um grupo de robôs móveis são o transporte de cargas, a busca e regaste de pessoas em escombros, o mapeamento de regiões e grandes áreas, a busca e desarmamento de minas terrestres, o futebol de robôs e a exploração espacial [26–34].

Dentre as abordagens de controle coordenado envolvendo cooperação de robôs móveis, três técnicas distintas podem ser mencionadas. São elas: formação líder-seguidor, método de estruturas virtuais e método baseado em comportamentos [35]. Na estrutura líder-seguidor, um robô nomeado líder é responsável por guiar todos os outros robôs que compõem a formação, a fim de levá-los a suas posições desejadas e fazê-los permanecer em formação durante todo o movimento. Normalmente, toda a responsabilidade sensorial e também a geração dos sinais de controle enviados a cada membro do pelotão é atribuída ao robô líder. Portanto, neste tipo de estrutura o líder é considerado o ente mais importante, pois sua falha certamente comprometeria a realização da tarefa programada [31, 32, 36, 37].

Tratando agora de uma estrutura virtual, o robô líder não é mais o ente principal, que passa a ser toda a formação, a qual é considerada um corpo rígido que se move para estabelecer uma figura geométrica pré-definida [33, 38, 39].

Por outro lado, em métodos baseados em comportamentos, um conjunto de procedimentos é definido para cada robô da equipe, tais como: buscar um alvo, evitar obstáculos e estabelecer a formação. Os sinais de controle enviados individualmente aos robôs correspondem, normalmente, a uma média ponderada de cada comportamento, aplicada a cada ente do pelotão [28, 40, 41].

Quanto à estratégia de controle, esta pode ser realizada de forma centralizada ou descentralizada [30]. Para o caso de um controle centralizado, o líder da formação é responsável por todas as informações referentes ao grupo, tais como posições relativas dos demais robôs e envio de sinais de controle para estabelecimento da formação [41]. Por outro lado, entende-se por controle descentralizado de formação aquele desprovido de tal unidade centralizadora de informação. Neste sentido, os robôs que constituem a formação possuem seus próprios sensores destinados a informar a posição atual no ambiente, o estado do espaço de trabalho, a posição corrente dos demais robôs e, por si só, constituir a formação [30, 32]. Entretanto, alguns pesquisadores denominam de formação descentralizada aquela na qual existe uma comunicação estruturada entre os robôs, isto é, a troca de informações referente à formação se propaga segundo um grafo pré-definido [35, 42]. Em [36], teoria de grafos é aplicada a uma formação descentralizada, na qual os robôs que a constituem podem ser tanto líder quanto seguidores. A mudança de estado é decorrente de falhas mecânicas do robô líder atual ou interrupção na troca de informações com os demais seguidores.

As vantagens e desvantagens do esquema de controle centralizado ou descentralizado têm sido objeto de discussão na comunidade científica. Sistemas centralizados de controle, por exemplo, podem gerir a informação global do ambiente e otimizar a coordenação entre os robôs ou o cumprimento da missão, reduzindo e/ou suavizando as trajetórias (ou caminhos) a serem percorridos. Além disso, eles podem facilmente identificar e gerenciar falhas de alguns dos robôs. Por outro lado, a unidade centralizadora de informação (normalmente o robô líder) se apresenta como fator limitante do sistema, no que se refere aos requisitos computacionais e de comunicação. Além disso, sua eventual falha compromete o sistema por completo. Por sua vez, sistemas descentralizados de controle não apresentam uma restrição computacional rígida, pois cada agente executa seu controle de forma independente, mas com um objetivo global em comum. Além disso, tais sistemas são inerentemente mais estáveis e robustos, pois o cumprimento da tarefa pode ser obtido ainda que suceda uma falha (ou mal comportamento) em algum dos robôs da formação. Em contrapartida, a dificuldade na localização e mapeamento global dificulta a coordenação e o comportamento ótimo da formação durante a execução da missão [43, 44]. Neste sen-

tido, portanto, a escolha da estratégia de controle de formação fica a cargo do projetista, o qual deve analisar as características da tarefa a ser executada pelo grupo de robôs.

Em tal contexto, este capítulo apresenta uma proposta de controle descentralizado de formação segundo a abordagem líder-seguidor. A formação é composta por dois robôs tipo unicycle com tração diferencial (conforme pode ser visto na Figura 23), que não compartilham informações. É responsabilidade do seguidor, posicionar-se em uma estrutura linear em relação ao líder. A identificação da posição atual do robô líder (que transporta um padrão a ser reconhecido durante a navegação) é efetuada pelo robô seguidor, através das medidas provenientes de um sensor de varredura laser instalado a bordo do mesmo. Uma estratégia sensorial para determinação das variáveis de formação linear é aqui proposta e testada experimentalmente. De posse de tais variáveis, uma lei de controle projetada segundo a teoria de Lyapunov é proposta e aplicada ao robô seguidor (ênfatizando controle descentralizado), para que ele alcance a postura no espaço de trabalho semi-estruturado que garanta a estrutura da formação.

O texto está organizado como segue: a Seção 3.1 ilustra as variáveis de formação envolvidas na estrutura linear líder-seguidor e as equações de estado do sistema; posteriormente, a Seção 3.2 apresenta o projeto do controlador descentralizado de formação e sua análise de estabilidade segundo Lyapunov; em seguida, a Seção 3.3 mostra a estratégia utilizada para obtenção das variáveis de formação com base no sistema sensorial



Figura 23: Robôs móveis unicycle que compõem a formação linear.

a bordo do robô seguidor; na sequência, a Seção 3.4 resalta os resultados experimentais e as discussões pertinentes no que diz respeito às leis de controle de formação propostas; finalmente, a Seção 3.5 destaca algumas conclusões e sugestões para trabalhos futuros.

3.1 Variáveis de Formação e Equações de Estado

Nesta seção, são apresentadas as variáveis de formação envolvidas no controle de formação descentralizado líder-seguidor (Figura 24). O vetor $[\rho_{LF} \ \beta_{LF} \ \theta_{LF}]^T$ contém tais variáveis, que representam a distância entre os robôs líder e seguidor ($\rho_{LF} \in \mathbb{R}^+$), o ângulo que eles mantêm entre si ($\beta_{LF} \in (0^\circ, 180^\circ)$), e o erro de orientação entre eles ($\theta_{LF} \in [-90^\circ, 90^\circ]$). Tais variáveis são completamente determinadas através das medidas provenientes do sensor laser a bordo do robô seguidor, e são atualizadas a cada 100 ms, conforme discutido na Seção 3.3.

Algumas relações matemáticas podem ser observadas diretamente da Figura 24, tais como

$$\{x_{LF} = \rho_{LF} \cos \beta_{LF}, \quad y_{LF} = \rho_{LF} \sin \beta_{LF}\}. \quad (3.1)$$

Ainda visualizando a Figura 24, é possível encontrar relações entre as variáveis de

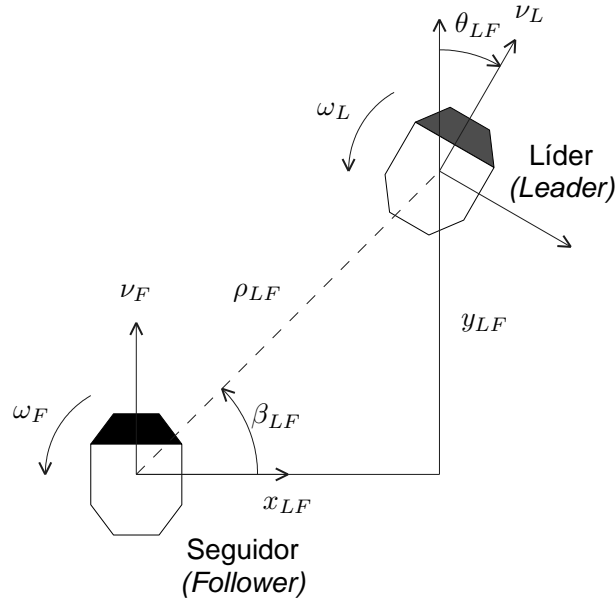


Figura 24: Variáveis de formação.

formação e as velocidades dos robôs nas direções x e y , dadas por

$$\begin{aligned}\dot{x}_{LF} &= -\nu_L \sin \theta_{LF} + \rho_{LF} \omega_F \sin \beta_{LF} \\ \dot{y}_{LF} &= \nu_L \cos \theta_{LF} - \rho_{LF} \omega_F \cos \beta_{LF} - \nu_F,\end{aligned}\tag{3.2}$$

tendo em mente que o sistema referencial é tomado em relação ao robô seguidor. Considerando o triângulo aparente em tal figura, temos que

$$\rho_{LF}^2 = x_{LF}^2 + y_{LF}^2,$$

cuja derivada temporal é dada por

$$\rho_{LF} \dot{\rho}_{LF} = x_{LF} \dot{x}_{LF} + y_{LF} \dot{y}_{LF}.$$

Substituindo estes resultados da Equação 3.2, obtém-se

$$\dot{\rho}_{LF} = \nu_L \sin(\beta_{LF} - \theta_{LF}) - \nu_F \sin \beta_{LF},\tag{3.3}$$

que é a equação que caracteriza a evolução temporal de ρ_{LF} .

Tomando a derivada temporal da relação trigonométrica

$$\tan \beta_{LF} = \frac{y_{LF}}{x_{LF}},$$

que resulta inicialmente em

$$\sec^2 \beta_{LF} \dot{\beta}_{LF} = \frac{x_{LF} \dot{y}_{LF} - y_{LF} \dot{x}_{LF}}{x_{LF}^2},$$

e utilizando a Equação 3.2, tem-se que

$$\dot{\beta}_{LF} = \frac{1}{\rho_{LF}} [\nu_L \cos(\beta_{LF} - \theta_{LF}) - \nu_F \cos \beta_{LF}] - \omega_F.\tag{3.4}$$

Por fim, a variação do erro angular entre os robôs líder e seguidor depende somente da diferença entre as velocidades angulares destes robôs. Logo, pode-se escrever que

$$\dot{\theta}_{LF} = \omega_L - \omega_F.\tag{3.5}$$

Portanto, as Equações 3.3, 3.4 e 3.5 descrevem o sistema de equações de estado do

robô seguidor com base nas variáveis de formação. Isto é,

$$\begin{cases} \dot{\rho}_{LF} &= \nu_L \sin(\beta_{LF} - \theta_{LF}) - \nu_F \sin \beta_{LF} \\ \dot{\beta}_{LF} &= \frac{1}{\rho_{LF}} [\nu_L \cos(\beta_{LF} - \theta_{LF}) - \nu_F \cos \beta_{LF}] - \omega_F \\ \dot{\theta}_{LF} &= \omega_L - \omega_F. \end{cases} \quad (3.6)$$

3.2 Controle Descentralizado de Formação

Os controladores propostos nesta seção também foram apresentados em [32] e [45] para estabelecer a formação líder-seguidor utilizando sensoriamento laser. O objetivo deste controlador é fazer com que $\rho_{LF} \rightarrow \rho_{LFd}$ e $\beta_{LF} \rightarrow \beta_{LFd}$ para $t \rightarrow \infty$, enquanto θ_{LF} deve tender a um valor limitado.

3.2.1 Abordagem por Lyapunov

Visando projetar um controlador com base na teoria de Lyapunov, definiram-se os erros das variáveis de formação a serem controlados. São eles

$$\begin{aligned} \tilde{\rho}_{LF} &= \rho_{LFd} - \rho_{LF} \\ \tilde{\beta}_{LF} &= \beta_{LFd} - \beta_{LF}. \end{aligned} \quad (3.7)$$

Tomando a derivada temporal de tais erros e sabendo que os valores de ρ_{LFd} e β_{LFd} são constantes, tem-se que

$$\begin{aligned} \dot{\tilde{\rho}}_{LF} &= -\dot{\rho}_{LF} \\ \dot{\tilde{\beta}}_{LF} &= -\dot{\beta}_{LF}. \end{aligned} \quad (3.8)$$

Propondo uma função candidata de Lyapunov definida positiva que represente a energia do erro das variáveis de formação controláveis, a saber,

$$V(\tilde{\rho}_{LF}, \tilde{\beta}_{LF}) = \frac{1}{2} \tilde{\rho}_{LF}^2 + \frac{1}{2} \tilde{\beta}_{LF}^2, \quad (3.9)$$

tomando sua derivada temporal e usando as Equações 3.3 e 3.4, obtém-se

$$\begin{aligned} \dot{V}(\tilde{\rho}_{LF}, \tilde{\beta}_{LF}) &= \tilde{\rho}_{LF} (\nu_F \sin \beta_{LF} - \nu_L \sin(\beta_{LF} - \theta_{LF})) \\ &\quad + \tilde{\beta}_{LF} \left(\frac{\nu_F \cos \beta_{LF} - \nu_L \cos(\beta_{LF} - \theta_{LF})}{\rho_{LF}} + \omega_F \right). \end{aligned} \quad (3.10)$$

Visando garantir que a derivada da função candidata de Lyapunov (Equação 3.10)

seja definida negativa para qualquer valor de $\tilde{\rho}_{LF}$ e $\tilde{\beta}_{LF}$, são propostos os sinais de controle

$$\nu_F = \frac{\nu_L \sin(\beta_{LF} - \theta_{LF}) - k_{L1} \tanh \tilde{\rho}_{LF}}{\sin \beta_{LF}} \quad (3.11)$$

$$\omega_F = \frac{\nu_L \cos(\beta_{LF} - \theta_{LF}) - \nu_F \cos \beta_{LF}}{\rho_{LF}} - k_{L2} \tanh \tilde{\beta}_{LF}, \quad (3.12)$$

onde k_{L1} e k_{L2} são constantes reais e positivas. Substituindo as Equações 3.11 e 3.12 em 3.10, resulta

$$\dot{V}(\tilde{\rho}_{LF}, \tilde{\beta}_{LF}) = -k_{L1} \tilde{\rho}_{LF} \tanh \tilde{\rho}_{LF} - k_{L2} \tilde{\beta}_{LF} \tanh \tilde{\beta}_{LF} < 0. \quad (3.13)$$

Conforme desejado, a derivada da função candidata de Lyapunov (Equação 3.13) é definida negativa, constatando, portanto, que $\tilde{\rho}, \tilde{\beta} \in L_\infty^n \cap L_2^n$.

As equações dos estados controláveis (Equações 3.3 e 3.4) em função das variáveis de erro são

$$\begin{cases} \dot{\tilde{\rho}}_{LF} = \nu_F \sin \beta_{LF} - \nu_L \sin(\beta_{LF} - \theta_{LF}) \\ \dot{\tilde{\beta}}_{LF} = \frac{1}{\rho_{LF}} [\nu_F \cos \beta_{LF} - \nu_L \cos(\beta_{LF} - \theta_{LF})] + \omega_F. \end{cases} \quad (3.14)$$

Em malha fechada, obtém-se

$$\dot{\tilde{\rho}}_{LF} + k_{L1} \tanh \tilde{\rho}_{LF} = 0 \quad (3.15)$$

$$\dot{\tilde{\beta}}_{LF} + k_{L2} \tanh \tilde{\beta}_{LF} = 0, \quad (3.16)$$

cujo ponto de equilíbrio é $[\tilde{\rho}_{LF} \ \tilde{\beta}_{LF}]^T = [0 \ 0]^T$, o qual é assintoticamente estável conforme conclusão tirada da Equação 3.13. Por conseguinte, tem-se que $\tilde{\rho}_{LF} \rightarrow 0$ e $\tilde{\beta}_{LF} \rightarrow 0$ para $t \rightarrow \infty$, ou ainda, $\rho_{LF} \rightarrow \rho_{LFd}$ e $\beta_{LF} \rightarrow \beta_{LFd}$ para $t \rightarrow \infty$.

Para fins de implementação, a velocidade angular do seguidor ω_F não necessita de uma dependência direta de sua própria velocidade linear ν_F . No entanto, substituindo a Equação 3.11 em 3.12, tem-se a nova definição dos sinais de controle enviados ao robô seguidor da formação, que são

$$\nu_F = \frac{\nu_L \sin(\beta_{LF} - \theta_{LF}) - k_{L1} \tanh \tilde{\rho}_{LF}}{\sin \beta_{LF}} \quad (3.17)$$

$$\omega_F = \frac{\nu_L \sin \theta_{LF} + k_{L1} \tanh \tilde{\rho}_{LF} \cos \beta_{LF}}{\rho_{LF} \sin \beta_{LF}} - k_{L2} \tanh \tilde{\beta}_{LF}. \quad (3.18)$$

Para finalizar esta seção, é importante comentar duas características associadas à lei de controle proposta. A primeira diz respeito à velocidade do robô líder, que deve ser estimada pelo seguidor durante a navegação (ver Equações 3.17 e 3.18). A segunda se

refere às duas singularidades do controlador proposto. A primeira destaca que a distância entre os robôs não deve ser igual a zero, $\rho_{LF} = 0$. Esta é uma singularidade perfeitamente respeitada na estrutura líder-seguidor, pois um robô não pode estar sobreposto ao outro. A segunda diz respeito aos valores de $\beta_{LF} = n\pi$, que significa que a formação não pode ser realizada com robôs alinhados lateralmente. Esta singularidade é causada pela limitação sensorial do robô seguidor, pois o sensor laser a bordo do mesmo visualiza somente a parte frontal de seu espaço de trabalho.

Um último aspecto deve ser destacado, que diz respeito ao erro de orientação entre os robôs θ_{LF} . Para que ocorra uma mudança nesta variável, um dos robôs da formação deve girar (observar Figura 24). Se o robô seguidor girar, irá ocorrer uma modificação na variável β_{LF} . Porém, sabe-se que em estado estacionário $\beta_{LF} \rightarrow \beta_{LFd}$. Portanto, θ_{LF} não será modificada por meio desta variável de formação, pois isto significaria retirar a formação da condição de equilíbrio. Por sua vez, uma rotação do robô líder poderia provocar uma variação de θ_{LF} . Entretanto, o controlador de posição proposto na Subseção 1.3.1, implementado neste robô, apresenta um comportamento assintoticamente estável, isto é, as variáveis envolvidas são bem comportadas, e dentre elas se inclui a orientação do robô líder. Logo, pode-se garantir, fisicamente, que θ_{LF} será limitado no equilíbrio, ou seja, $\theta_{LF} \in L_\infty^n$.

3.2.2 Abordagem por Cinemática Inversa

Esta subseção aborda uma proposta de projeto de controlador baseado na cinemática inversa para o problema de controle de formação linear líder-seguidor apresentado na Seção 3.1 e discutido inicialmente na Subseção 3.2.1.

Primeiramente, deve-se colocar as equações de estado do sistema sob a forma

$$\dot{\mathbf{x}} = g(\mathbf{x})\mathbf{u} + h(\mathbf{x}). \quad (3.19)$$

Assumindo $\mathbf{x} = [\rho_{LF} \ \beta_{LF}]^T$, a modelagem por cinemática inversa é dada por

$$\underbrace{\begin{bmatrix} \dot{\rho}_{LF} \\ \dot{\beta}_{LF} \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} -\sin\beta_{LF} & 0 \\ -\frac{\cos\beta_{LF}}{\rho_{LF}} & -1 \end{bmatrix}}_{g(\mathbf{x})} \underbrace{\begin{bmatrix} \nu_F \\ \omega_F \end{bmatrix}}_{\mathbf{u}} + \underbrace{\begin{bmatrix} \nu_L \sin(\beta_{LF} - \theta_{LF}) \\ \frac{\nu_L \cos(\beta_{LF} - \theta_{LF})}{\rho_{LF}} \end{bmatrix}}_{h(\mathbf{x})}, \quad (3.20)$$

onde \mathbf{x} representa as variáveis de formação controláveis e \mathbf{u} representa os sinais de controle. O objetivo em questão é encontrar um sinal de controle \mathbf{u} que faça com que $\mathbf{x} \rightarrow \mathbf{x}_d$ para

$t \rightarrow \infty$. Considerando

$$\mathbf{u} = g^{-1}(\mathbf{x})(\eta - h(\mathbf{x})), \quad (3.21)$$

supondo que $g(\mathbf{x})$ seja inversível, e assumindo que

$$\eta = \dot{\mathbf{x}}_d + \mathbf{K}f(\tilde{\mathbf{x}}), \quad (3.22)$$

onde \mathbf{K} é uma matriz de ganhos definida positiva, tem-se, em malha fechada, que

$$\dot{\mathbf{x}} = g(\mathbf{x}) [g^{-1}(\mathbf{x})(\eta - h(\mathbf{x}))] + h(\mathbf{x})$$

$$\dot{\mathbf{x}} = [\dot{\mathbf{x}}_d + \mathbf{K}f(\tilde{\mathbf{x}}) - h(\mathbf{x})] + h(\mathbf{x})$$

$$\dot{\mathbf{x}} = \dot{\mathbf{x}}_d + \mathbf{K}f(\tilde{\mathbf{x}}),$$

e, por fim,

$$\dot{\tilde{\mathbf{x}}} + \mathbf{K}f(\tilde{\mathbf{x}}) = 0. \quad (3.23)$$

A fim de analisar a estabilidade do equilíbrio correspondente à Equação 3.23, aplica-se a teoria de Lyapunov para sistemas lineares. Para isto, propõe-se a função candidata de Lyapunov

$$V(\tilde{\mathbf{x}}) = \frac{1}{2} \tilde{\mathbf{x}}^T \tilde{\mathbf{x}},$$

cujas derivada temporal, usando a Equação 3.23, é dada por

$$\begin{aligned} \dot{V}(\tilde{\mathbf{x}}) &= \tilde{\mathbf{x}}^T \dot{\tilde{\mathbf{x}}} \\ &= -\tilde{\mathbf{x}}^T \mathbf{K}f(\tilde{\mathbf{x}}). \end{aligned} \quad (3.24)$$

Assumindo $f(\tilde{\mathbf{x}}) = \begin{bmatrix} \tanh \tilde{\rho}_{LF} \\ \tanh \tilde{\beta}_{LF} \end{bmatrix}$ e $\mathbf{K} = \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix}$, sendo $k_1 > 0$ e $k_2 > 0$, é possível afirmar que

$$\dot{V}(\tilde{\mathbf{x}}) = -\tilde{\mathbf{x}}^T \mathbf{K}f(\tilde{\mathbf{x}}) < 0, \quad (3.25)$$

isto é, \dot{V} é definida negativa. Logo, $\tilde{\mathbf{x}} \in L_\infty^n \cap L_2^n$, o que implica que $\tilde{\mathbf{x}} \rightarrow 0$ para $t \rightarrow \infty$, ou ainda, $\mathbf{x} \rightarrow \mathbf{x}_d$ com $t \rightarrow \infty$.

Tendo em vista a estratégia de controle de formação abordada nesta Dissertação, $\mathbf{x}_d = [\rho_{LFd} \ \beta_{LFd}]^T$ é um vetor constante, e logo, $\dot{\mathbf{x}}_d = [0 \ 0]^T$. Assim sendo, o vetor dos sinais de controle dado pela Equação 3.21 torna-se

$$\mathbf{u} = \begin{bmatrix} \nu_F \\ \omega_F \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{-1}{\sin \beta_{LF}} & 0 \\ \frac{\cos \beta_{LF}}{\rho_{LF} \sin \beta_{LF}} & -1 \end{bmatrix}}_{g^{-1}(\mathbf{x})} \left(\begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \begin{bmatrix} \tanh \tilde{\rho}_{LF} \\ \tanh \tilde{\beta}_{LF} \end{bmatrix} - \begin{bmatrix} \nu_L \sin(\beta_{LF} - \theta_{LF}) \\ \frac{\nu_L \cos(\beta_{LF} - \theta_{LF})}{\rho_{LF}} \end{bmatrix} \right). \quad (3.26)$$

Ao calcular a matriz inversa de $g(\mathbf{x})$ para obter \mathbf{u} (Equação 3.26), aparece de forma explícita a singularidade para valores de $\beta_{LF} = n\pi$, para $n \in \mathbb{N}$. A presença desta singularidade se deve às características do sistema sensorial adotado, conforme mencionado nas considerações finais da Subseção 3.2.1. A singularidade referente à sobreposição dos robôs da formação ($\rho_{LF} = 0$) também afeta a inversão da matriz e, por conseguinte, o cálculo dos sinais de controle do robô seguidor. Entretanto, esta última é perfeitamente respeitada, no sistema de controle proposto.

Reescrevendo a Equação 3.26 a fim de colocá-la em uma forma mais apresentável, tem-se os sinais de controle enviados ao robô seguidor da formação dados por

$$\mathbf{u} = \begin{bmatrix} \nu_F \\ \omega_F \end{bmatrix} = \begin{bmatrix} \frac{\nu_L \sin(\beta_{LF} - \theta_{LF}) - k_1 \tanh \rho_{LF} \tilde{\rho}_{LF}}{\sin \beta_{LF}} \\ \frac{\nu_L \sin \theta_{LF} + k_1 \tanh \tilde{\rho}_{LF} \cos \beta_{LF}}{\rho_{LF} \sin \beta_{LF}} - k_2 \tanh \tilde{\beta}_{LF} \end{bmatrix}. \quad (3.27)$$

Como é possível verificar, para tal problema de controle de formação, o desenvolvimento por Cinemática Inversa fornece sinais de controle idênticos àqueles utilizados no projeto de controlador baseado na Teoria de Lyapunov (vide Equações 3.17 e 3.18).

3.3 Estratégia Sensorial

3.3.1 Cálculo das Variáveis de Formação

Esta subseção apresenta uma proposta de estimativa dos valores das variáveis de formação $[\rho_{LF} \ \beta_{LF} \ \theta_{LF}]^T$ a partir das medidas da varredura do sensor laser instalado a bordo do robô seguidor (Figura 25).

Inicialmente, monta-se sobre o robô líder da formação um semi-cilindro de diâmetro igual a $200mm$ para sua identificação no ambiente (vide Figura 26(a)). Algumas configurações de posição e orientação deste padrão relativas ao robô seguidor podem ser vistas nas Figuras 26(b), 27(a) e 27(b). Em seguida, por métodos geométricos são obtidas as equações que regem o cálculo das variáveis de formação. Desta forma, é possível obter

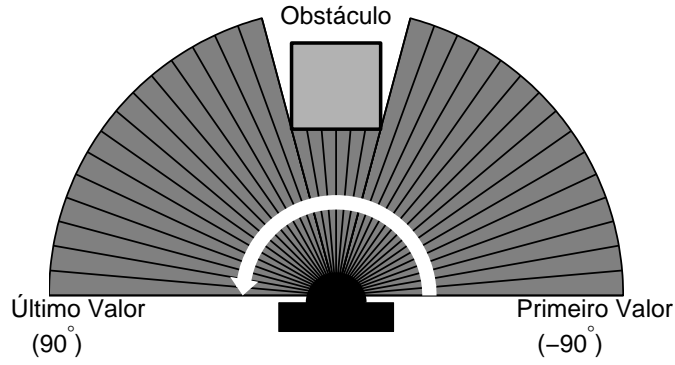


Figura 25: Sensor de varredura laser utilizado na formação líder-seguidor.

o estado corrente da estrutura líder-seguidor sem a necessidade de compartilhamento de informações entre os robôs (controle descentralizado). Os sistemas de coordenadas adotados para o robô líder e para o robô seguidor são referenciados, respectivamente, por $\langle L \rangle$ e $\langle F \rangle$.

A Figura 26(b) ilustra as medidas de distância entre o sensor laser montado no robô seguidor e o início e fim do padrão *visto* por este sensor, representadas por ρ_1 e ρ_2 , as quais são tomadas, respectivamente, nos ângulos β_1 e β_2 da varredura laser. São também apresentadas as variáveis de formação controláveis ρ_{LF} e β_{LF} , cujos valores devem ser estimados.

Na Figura 27(a), é definido o padrão virtual p_v , cuja dimensão, em função da pose relativa entre os robôs, é sempre menor ou igual à dimensão do padrão real p_r (vide

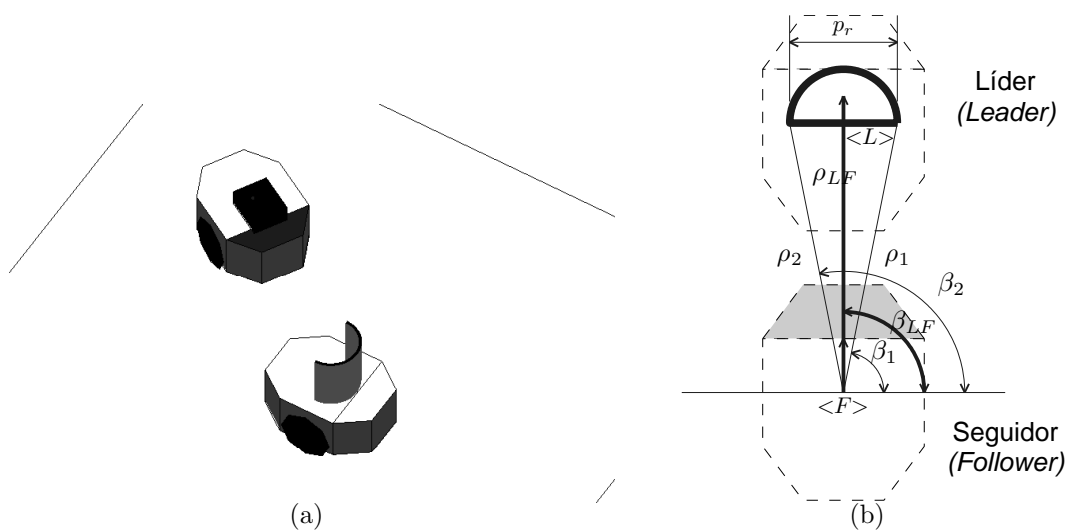


Figura 26: O padrão a bordo do robô líder. (a) Ambiente de simulação da formação líder-seguidor. (b) Distância ρ_{LF} entre os robôs em função das medidas de laser.

Figura 26(b)). Assim sendo, o padrão virtual é dado por

$$p_v^2 = \rho_1^2 + \rho_2^2 - 2\rho_1\rho_2 \cos \beta_*, \quad (3.28)$$

onde $\beta_* = \beta_2 - \beta_1$. A razão entre as dimensões do padrão virtual e do padrão real resulta em

$$\cos \gamma_1 = \frac{p_v}{p_r}. \quad (3.29)$$

O máximo de γ_1 é obtido quando $p_v = p_r/\sqrt{2}$, situação que ocorre quando se tem um quadrado inscrito numa circunferência, a qual é representada aqui pelo padrão. Portanto, pode-se verificar que $\gamma_1 \in [-45^\circ, 45^\circ]$.

Na Figura 27(b), são apresentadas as variáveis θ_{LF} e γ_2 , que indicam, respectivamente, o erro de orientação entre o robô seguidor e o líder e o ângulo de rotação aparente do padrão, conforme este é *visto* pelo sensor laser. O valor de γ_2 , analisando as Figuras 26(b) e 27(b), é dado por

$$\tan \gamma_2 = \frac{\rho_1 \sin \beta_1 - \rho_2 \sin \beta_2}{\rho_1 \cos \beta_1 - \rho_2 \cos \beta_2}. \quad (3.30)$$

De posse de γ_1 e γ_2 , o erro de orientação da formação líder-seguidor é dada por

$$\theta_{LF} = \begin{cases} \text{sign}(\gamma_2)(\gamma_1 - |\gamma_2|) & \text{se } \rho_1 > \rho_2 \text{ e } \gamma_2 > 0 \text{ ou } \rho_1 < \rho_2 \text{ e } \gamma_2 < 0, \\ \text{sign}(\gamma_2)(\gamma_1 + |\gamma_2|) & \text{caso contrário.} \end{cases} \quad (3.31)$$

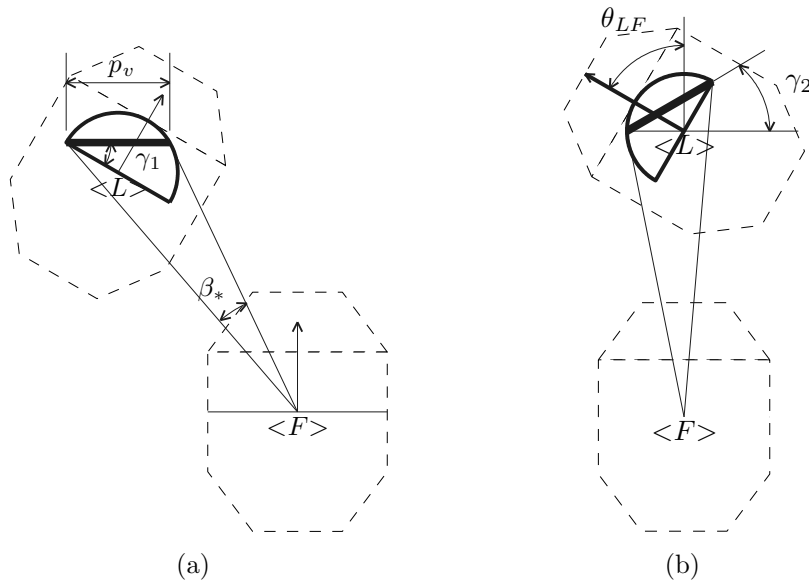


Figura 27: Padrão montado sobre o robô líder. (a) Padrão virtual (padrão real como é *visto* pelo robô seguidor). (b) Definição do erro de orientação θ_{LF} e do ângulo de rotação aparente do padrão γ_2 .

O ângulo da formação líder-seguidor, β_{LF} , representa o ângulo entre o eixo de referência do robô seguidor $\langle F \rangle$ e o vetor de distância ρ_{LF} que une ambos os robôs. Da Figura 28 obtém-se que

$$\beta_{LF} = 90^\circ + \alpha_F, \quad (3.32)$$

onde $\alpha_F = \theta_F - \psi_F$, sendo que ψ_F é a orientação do robô seguidor com relação ao referencial inercial $\langle g \rangle$ e θ_F é o erro de orientação entre este referencial e o vetor ρ_{LF} que une ambos robôs. Em outra palavras, $\langle L \rangle$ se comporta como um alvo móvel para o robô seguidor.

Uma vez obtidas as variáveis de formação supracitadas, tem-se, por fim, que

$$\rho_{LF}^2 = \begin{cases} \rho_1^2 + \frac{p_r^2}{4}, & \text{se } \beta_1 \geq 90^\circ, \beta_2 > 90^\circ \text{ e } |\theta_{LF}| < 90^\circ - \beta_{LF} \\ & \text{ou } \beta_1 < 90^\circ, \beta_2 \leq 90^\circ \text{ e } |\theta_{LF}| \geq 90^\circ - \beta_{LF}, \\ \rho_2^2 + \frac{p_r^2}{4}, & \text{se } \beta_1 < 90^\circ, \beta_2 \leq 90^\circ \text{ e } |\theta_{LF}| < 90^\circ - \beta_{LF} \\ & \text{ou } \beta_1 \geq 90^\circ, \beta_2 > 90^\circ \text{ e } |\theta_{LF}| \geq 90^\circ - \beta_{LF}, \\ \left(\rho_1 \cos \beta_1 - \frac{p_r}{2} \right)^2 + \rho_1^2 \sin^2 \beta_1, & \text{se } \beta_1 < 90^\circ, \beta_2 > 90^\circ \text{ e } \theta_{LF} = 0^\circ. \end{cases} \quad (3.33)$$

Entretanto, para obter os valores de β_{LF} e, por conseguinte, de ρ_{LF} , é essencial conhecer os valores de ψ_F e θ_F . Porém, para obtê-los, é necessário utilizar a odometria do robô seguidor. Esta, por sua vez, fornece valores não muito confiáveis, devido à acumulação de erros durante a navegação. Neste sentido, propõe-se um modelo simplificado para

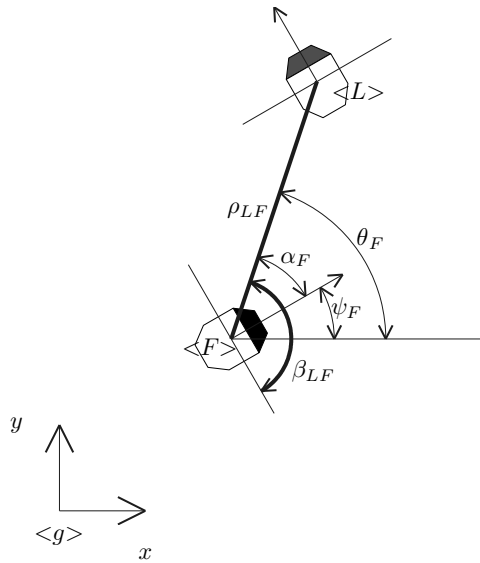


Figura 28: Obtendo o ângulo β_{LF} através da odometria do robô seguidor.

determinar o valor de β_{LF} , o qual é calculado agora por

$$\beta_{LF} \approx \frac{\beta_1 + \beta_2}{2}. \quad (3.34)$$

Tendo em vista que ruídos nas medidas da varredura laser podem introduzir um grande erro no cálculo da dimensão do padrão virtual, devido à grande discrepância métrica entre a distância entre os robôs e tal dimensão, e sabendo que a varredura angular do sensor utilizado é discreta (intervalos de 1°), é adotada a aproximação

$$\theta_{LF} \approx \gamma_2, \quad (3.35)$$

ao invés da Equação 3.31. Neste caso, torna-se mais importante saber o sentido de orientação do robô líder perante o seguidor, e não o valor exato desta.

Finalmente, tendo os valores aproximados de θ_{LF} e β_{LF} , o valor de ρ_{LF} pode ser obtido pela Equação 3.33, ou aproximado por

$$\rho_{LF} \approx \mathbf{M} \angle \beta_{LF}, \quad (3.36)$$

onde \mathbf{M} é o vetor de medidas de distância fornecidas pelo sensor laser. Em outras palavras, tal equação diz que ρ_{LF} é aproximadamente a medida de distância obtida na posição angular β_{LF} do vetor \mathbf{M} .

3.3.2 Estratégia de Busca do Líder

Como o sensor laser oferece somente informação unidimensional, a determinação da posição e orientação do líder se faz através da busca do padrão supracitado (vide Figura 26(a)), o qual deve ser distinto dos outros objetos no ambiente semi-estruturado. A Figura 29 ilustra um perfil de distâncias em função do ângulo de resolução do sensor durante a navegação de uma formação linear com $\rho_{LF} = 1m$ e $\beta_{LF} = 90^\circ$. Na parte inferior da Figura 29 é apresentada a diferença entre medidas angulares sucessivas do perfil, destacando suas descontinuidades. No que diz respeito ao padrão, tais descontinuidades sempre corresponderão a um pico negativo seguido de outro positivo, no que se refere à diferença entre distâncias sucessivas no vetor \mathbf{M} . Este é o fundamento de busca do robô líder.

A estratégia de busca consta em propor dois limiares, um positivo e outro negativo aplicados ao vetor de diferenças entre valores sucessivos do vetor \mathbf{M} . Conforme comentado anteriormente, o padrão gera descontinuidades no perfil laser, bem como os objetos

existentes no ambiente de navegação. Portanto, o intuito de propor tais limiares é simplesmente o de eliminar descontinuidades de baixa intensidade (ruídos), acelerando, assim, o processo de busca.

Sendo assim, inicialmente, faz-se uma busca pelos picos negativos existentes no perfil de diferenças entre medidas sucessivas, e se armazenam tais picos em um vetor auxiliar

$$\mathbf{N} = \begin{bmatrix} N_1 & N_2 & \cdots & N_n \end{bmatrix}, \text{ para } n < 180.$$

Em seguida, de modo similar, realiza-se a busca pelos picos positivos, que são armazenados em

$$\mathbf{P} = \begin{bmatrix} P_1 & P_2 & \cdots & P_p \end{bmatrix}, \text{ para } p < 180.$$

Posteriormente, verifica-se a condição $N_i < P_j$, para $i = 1, 2, \dots, n$ e $j = 1, 2, \dots, p$. Esta condição enfatiza a primeira premissa de busca do padrão, isto é, a posição angular em que ocorre o pico negativo deve ser menor que aquela onde ocorre o pico positivo. Se esta condição não for satisfeita, o processo de busca prossegue. Caso contrário, calcula-se a dimensão do padrão virtual p_v , visto que os valores de β_1 , β_2 , ρ_1 e ρ_2 são dados, respectivamente, por N_i , P_j , $\mathbf{M} \angle N_i$ e $\mathbf{M} \angle P_j$.

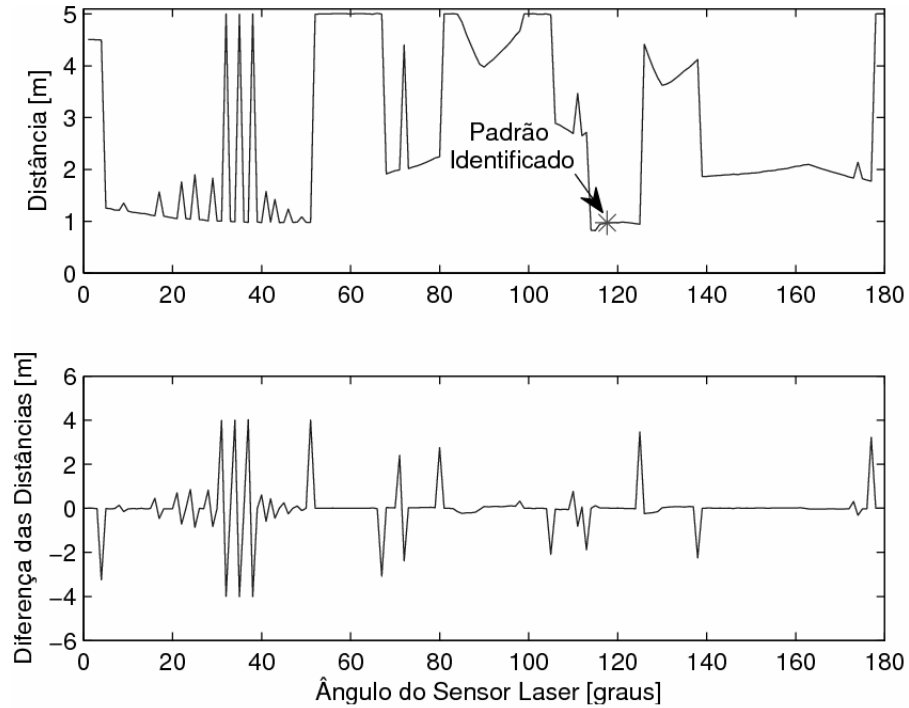


Figura 29: Perfil das medidas do sensor laser no ambiente e a diferença entre medidas de distância consecutivas.

A segunda premissa, diz respeito ao intervalo em que se encontra o padrão virtual p_v , isto é, $p_r/\sqrt{2} \leq p_v \leq p_r$. Assim, caso $p_v > p_r$, o processo de busca prossegue. Caso contrário, assume-se que um candidato a padrão foi encontrado, e este é armazenado. Porém, pode ocorrer a situação de se encontrar múltiplos candidatos a padrão no perfil de medidas fornecidas pelo sensor laser. Daí, considera-se eleito aquele candidato mais próximo ao robô seguidor.

Finalmente, conclui-se que as leituras fornecidas pelo sensor laser são suficientes para o controle de formação líder-seguidor. Afinal, os valores de ρ_1 , ρ_2 , β_1 e β_2 determinados nesta estratégia são suficientes para computar as variáveis de formação $[\rho_{LF}, \beta_{LF}, \theta_{LF}]^T$, conforme explicitado na Seção 3.3.1, e, por conseguinte, determinar a posição e orientação do robô líder.

3.4 Resultados e Discussão

Nesta seção são apresentados experimentos e simulações utilizando a proposta de controle descentralizado de formação descrita neste capítulo. Dois robôs Pioneer 2DX (tipo unicycle) foram aqui utilizados. Foram assumidos os seguintes valores para as constantes envolvidas no controlador descentralizado de formação: $k_1 = 0,25m/s$ e $k_2 = 0,5rad/s$.

3.4.1 Experimento 1 - Estacionamento

O primeiro experimento tem como objetivo principal destacar a estratégia sensorial de busca do padrão. Neste caso, o robô líder está localizado na posição $(3000mm, -1200mm)$, com orientação igual a zero com respeito ao eixo x , e o mesmo permanece estático ($\nu_L = 0$). Por outro lado, o robô seguidor está localizado nas coordenadas $(0mm, 0mm)$, com orientação também igual a zero com respeito ao eixo x . A estrutura linear desejada é uma distância entre os robôs $\rho_{LFd} = 500mm$ e um ângulo de formação $\beta_{LFd} = 120^\circ$.

A Figura 30(a) ilustra a trajetória do robô seguidor para atingir e permanecer na formação linear desejada. O robô seguidor é plotado a cada $3s$, durante os primeiros $20s$ de experimento, e o caminho que ele percorre no plano \mathbf{XY} foi reconstruído a partir dos dados da sua odometria.

A evolução das variáveis de formação pode ser visualizada na Figura 30(b). Conforme esperado, as variáveis ρ_{LF} e β_{LF} tendem assintoticamente aos valores desejados para $t \rightarrow \infty$, seguindo, portanto, o conceito de estabilidade assintótica do controle

descentralizado de formação baseado na Teoria de Lyapunov. Adicionalmente, é possível verificar que θ_{LF} tende assintoticamente a um valor constante quando $t \rightarrow \infty$.

Os sinais de controle enviados e efetivamente executados pelo robô seguidor são apresentados na Figura 30(c). Como se pode verificar, ν_F e ω_F tendem assintoticamente a zero, tendo em vista a dependência direta dos erros das variáveis de formação.

A velocidade linear aproximada do robô líder ν_L pode ser extraída a partir da equação de estado que define a distância líder-seguidor (Equação 3.3). O valor de ν_L é dito aproximado pois é decorrente da discretização de tal equação, onde o período de amostragem adotado é $T = 100ms$, igual ao período de amostragem com que trabalham os robôs móveis utilizados. Após aproximação, ν_L é suavizado utilizando um Filtro de Kalman¹. O valor aproximado de ν_L obtido durante o experimento pode ser visto na Figura 30(c).

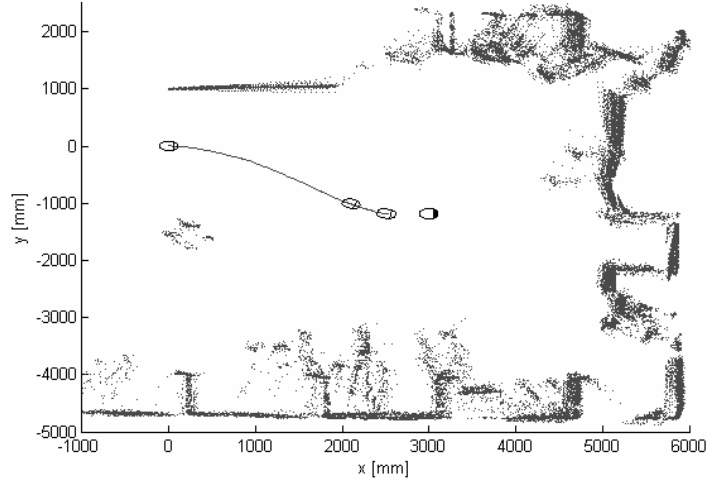
É importante mencionar que para determinação das variáveis de formação neste experimento, e nos demais subseqüentes nesta Dissertação, foram utilizadas as Equações 3.33 e 3.31 para determinar ρ_{LF} e θ_{LF} , respectivamente, e a aproximação apresentada na Equação 3.34 para determinar β_{LF} , eliminando, assim, a dependência da odometria do robô seguidor na estimação do estado da formação. A fim de suavizar os valores obtidos para as variáveis de formação, foram usados filtros de Kalman unidimensionais.

3.4.2 Experimento 2 - Busca de um alvo

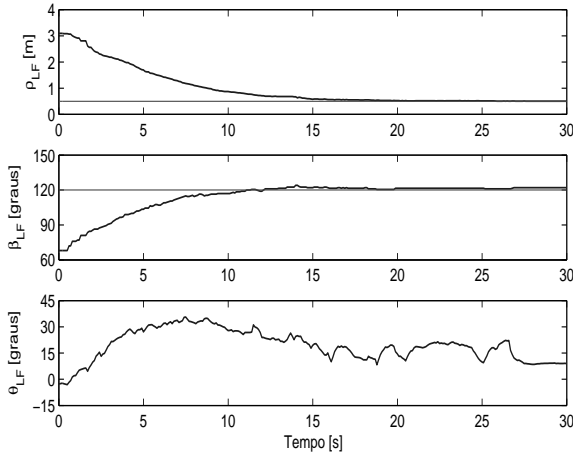
No segundo experimento realizado, o robô líder da formação tem implementado o controlador de posição descrito na Seção 1.3.1, cujo objetivo, neste caso, é alcançar o alvo localizado nas coordenadas $(3000mm, -1200mm)$, sendo que ele parte inicialmente da posição $(0mm, 0mm)$. Por outro lado, o robô seguidor, inicialmente posicionado nas coordenadas $(-1500mm, 0mm)$, deve buscar o robô líder utilizando a estratégia sensorial demonstrada anteriormente (Seção 3.3), e, por fim, atingir a formação desejada, aqui estabelecida por $\rho_{LFd} = 0,5m$ e $\beta_{LFd} = 60^\circ$.

A Figura 31(a) ilustra a trajetória descrita pela formação líder-seguidor durante a execução da tarefa. A postura do líder (plotado na cor preta) é aquela que é *vista* pelo robô seguidor (cor cinza), conforme estado atual das variáveis de formação. Pode-se verificar que o robô líder da formação atinge o alvo, conforme previsto na prova de estabilidade do controlador de posição. Os sinais de controle enviado e efetivamente desenvolvidos pelo robô líder para atingir o alvo são mostrados na Figura 31(c). É importante ressaltar

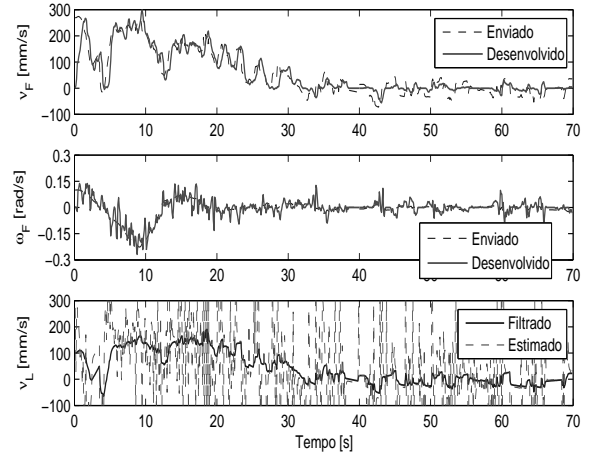
¹Detalhes sobre Filtro de Kalman podem ser vistos no Anexo 1.



(a) Caminho percorrido pelo robô seguidor para alcançar a postura desejada e estabelecer a formação.



(b) Evolução das variáveis de formação.



(c) Sinais de controle enviados e efetivamente desenvolvidos pelo robô seguidor durante a execução da tarefa de estacionamento, e velocidade linear ν_L estimada do robô líder.

Figura 30: Experimento 1 - Estacionamento.

que tais sinais tendem assintoticamente a zero, devido à dependência direta dos erros das variáveis de controle. Portanto, conclui-se experimentalmente que o robô permanece estático após atingir o alvo. A Figura 31(b) destaca os erros de posição (ρ_L) e orientação (α_L) e a convergência assintótica dos mesmos a zero. Vale lembrar que os dados aqui mostrados foram extraídos da odometria do robô líder.

Adicionalmente, constata-se que o robô seguidor percorre um caminho que lhe permite estar em formação linear. A evolução das variáveis de formação pode ser observada na Figura 31(d). O comportamento assintótico das mesmas em direção aos valores desejados (para as variáveis controladas ρ_{LF} e β_{LF}) e a tendência a um valor limitado (para θ_{LF})

podem ser vistos em tal figura, comprovando, mais uma vez, a estabilidade do ponto de equilíbrio (Seção 3.2).

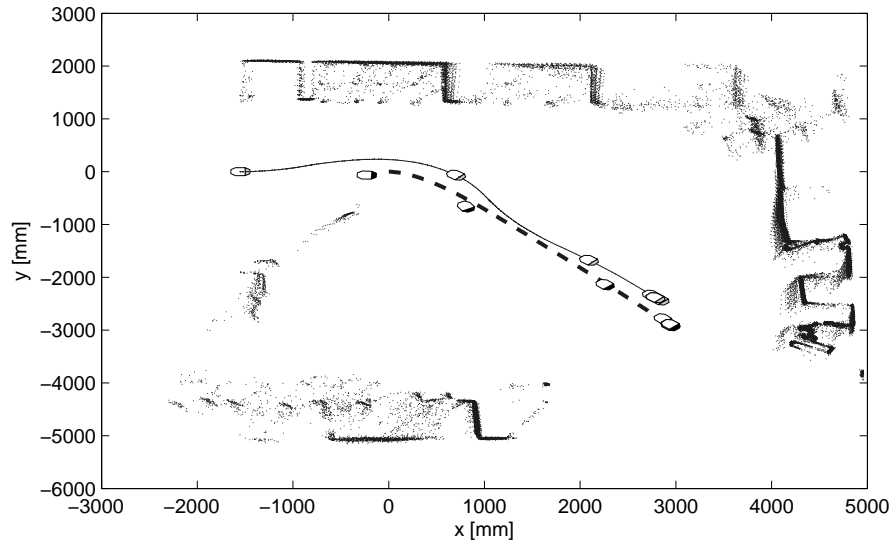
Os sinais de controle enviados e efetivamente desenvolvidos pelo robô seguidor para atingir e permanecer na formação são mostrados na Figura 31(e), onde também é apresentada a velocidade linear ν_L estimada do robô líder, essencial para a realização da tarefa de seguimento. É importante comentar que nos primeiros 5s, aproximadamente, o robô seguidor está se posicionando em relação ao robô líder. Quando próximo do valor desejado da formação, o robô líder passa a realizar a tarefa de posicionamento que lhe foi destinada, assim como o robô seguidor busca permanecer em formação.

3.4.3 Experimento 3 - Caminho circular

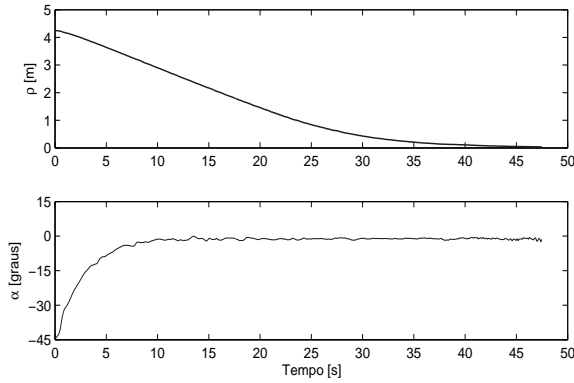
No terceiro experimento realizado neste capítulo, o objetivo é verificar a robustez do sistema de seguimento durante a execução de um caminho circular. Para o robô líder foram estipulados valores fixos de velocidades linear ($\nu_L = 0,15m/s$) e angular ($\omega_L = 0,1rad/s$), o que equivale a dizer que o mesmo executa um caminho circular (em malha aberta) de raio $r = 1,5m$ com uma velocidade tangencial $v = 0,15m/s$. A formação líder-seguidor desejada é caracterizada por $\rho_{LFd} = 0,7m$ e $\beta_{LFd} = 120^\circ$ nos primeiros 40s e $\beta_{LFd} = 90^\circ$ nos instantes finais do experimento. O robô líder está posicionado inicialmente nas coordenadas $(0mm, 0mm)$, com orientação igual a zero, enquanto o robô seguidor, fora de formação, está localizado nas coordenadas $(-700mm, -300mm)$.

A Figura 32(a) ilustra o caminho percorrido pelo robô seguidor durante a tarefa de seguimento. Em linha tracejada, é apresentado o caminho circular executado em malha aberta pelo robô líder. O robô seguidor é apresentado na figura com a parte frontal na cor cinza, enquanto o robô líder, conforme é visto pelo seguidor, é plotado na cor preta. A evolução das variáveis de formação é ilustrada na Figura 32(b). Vale destacar que o controlador proposto permanece estável após a alteração nos valores desejados (referências), conforme previsto, pois apenas os valores de referência são alterados. A convergência assintótica aos valores desejados também pode ser verificada.

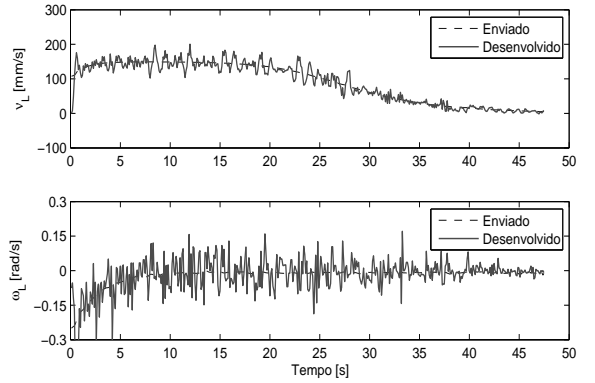
Os sinais de controle podem ser vistos na Figura 32(c), assim como a velocidade linear ν_L estimada do robô líder. As modificações das referências (β_{LFd}) refletem diretamente nos sinais de controle, causando variações bruscas nos mesmos, a fim de manter a formação no estado desejados. É fácil verificar que a velocidade linear ν_F do seguidor é inferior à do líder ν_L , quando o seguimento é realizado na parte interna da circunferência. Por outro lado, $\nu_F > \nu_L$ para o seguimento realizado pela parte externa da circunferência



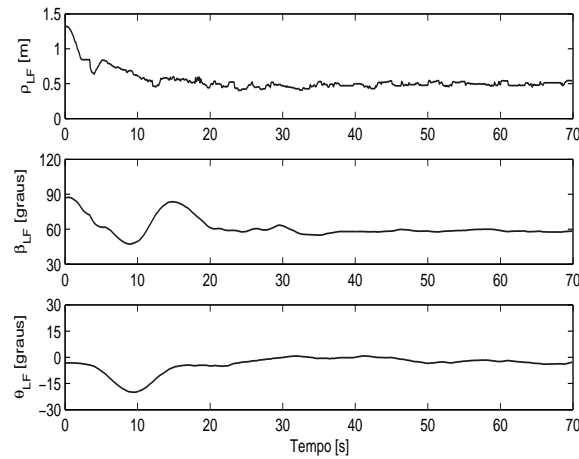
(a) Caminho percorrido pelo robô seguidor para alcançar a postura desejada e estabelecer a formação.



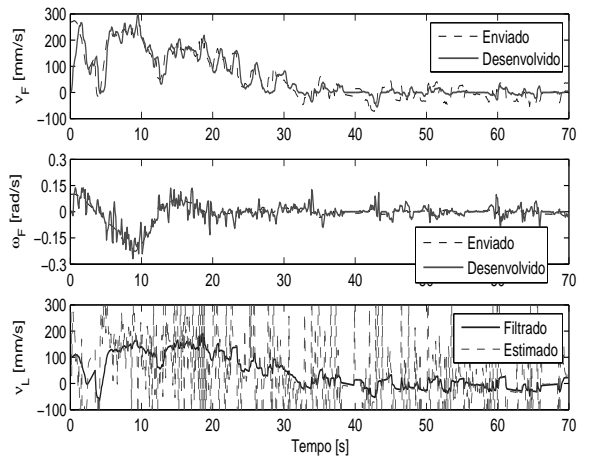
(b) Variáveis de erro do controlador de posição do líder.



(c) Sinais de controle enviados e efetivamente desenvolvidos pelo robô líder.



(d) Evolução das variáveis de formação.



(e) Sinais de controle enviados e efetivamente desenvolvidos pelo robô seguidor, e velocidade linear v_L estimada do robô líder.

Figura 31: Experimento 2 - Controle de Posição em Formação.

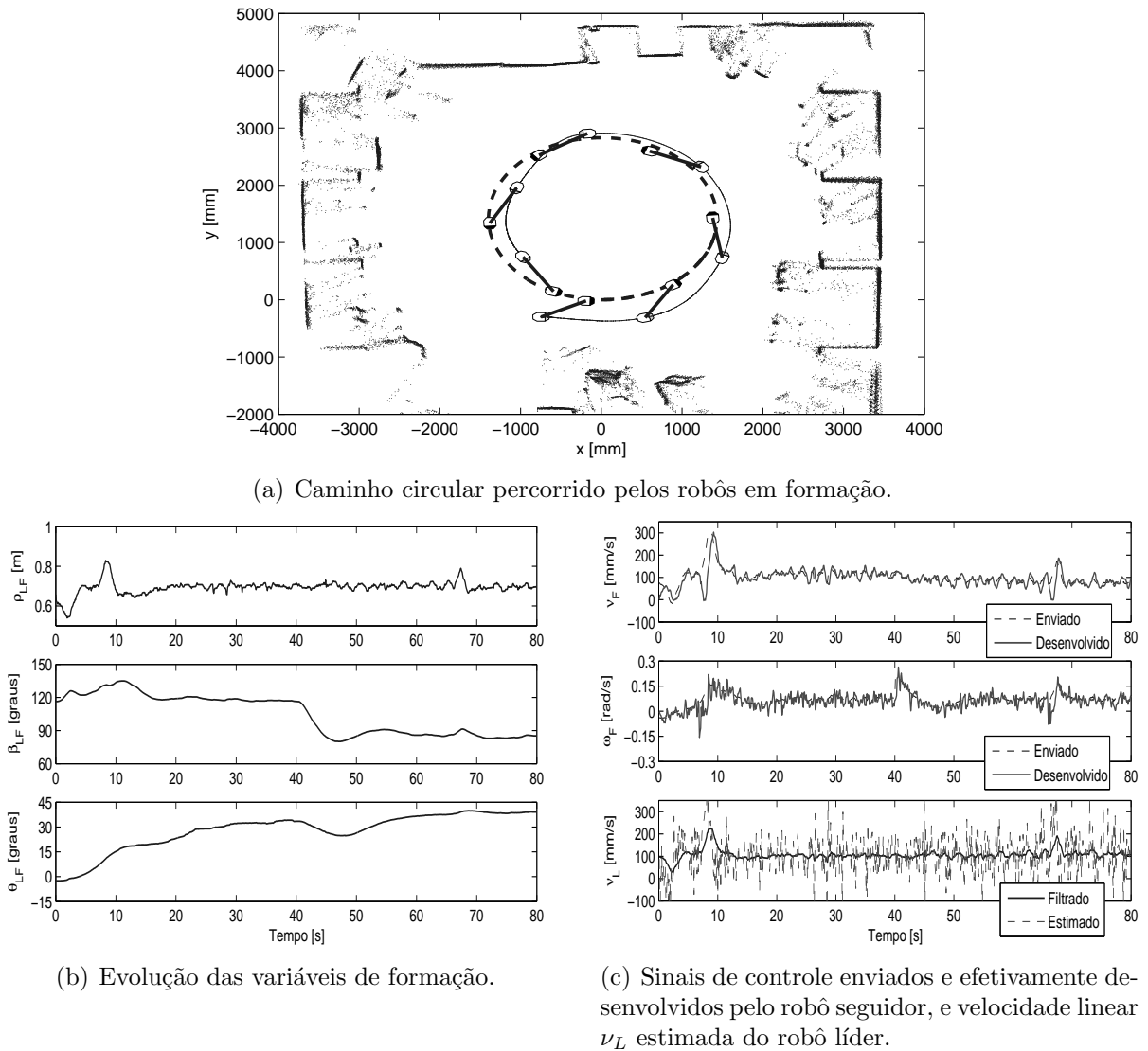


Figura 32: Experimento 3 - Caminho circular por mudança de referência do ângulo de formação.

traçada pelo robô líder. Tais diferenças de velocidades resultam em velocidades de escape tangencial (centrífugas) distintas, o que ocasiona em um erro no estado estacionário da formação, conforme pode ser visto durante a evolução das variáveis de formação.

3.5 Conclusões

Através do controlador proposto neste capítulo é possível comprovar, experimentalmente, a convergência assintótica das variáveis de formação controladas (ρ_{LF} e β_{LF}) para seus valores desejados, resultando, assim, na formação linear líder-seguidor desejada. Adicionalmente, é possível constatar que o erro de orientação líder-seguidor, θ_{LF} , tende a um

valor limitado, em regime estacionário.

Mediante a estratégia de busca do robô líder, constatou-se que é possível encontrar os valores reais (ou aproximados) das variáveis de formação envolvidas na estrutura líder-seguidor, desde que se tenha um sensor adequado a bordo do robô seguidor e um padrão característico montado no robô líder. No caso deste trabalho foi utilizado um sensor laser de varredura, com horizonte de varredura de 180° . Pode-se, porém, usar outro tipo de sensor, desde que o mesmo permita estimar as variáveis ρ_{LF} e β_{LF} , as quais determinam a estrutura da formação.

É possível verificar nos experimentos realizados na Seção 3.4 que há diferenças entre os sinais de controle enviados e efetivamente executados. A presença de oscilações sobrepostas aos valores desejados destes sinais se deve basicamente à dinâmica não modelada dos robôs Pioneer. Pode-se mencionar, por exemplo, a massa adicionada ao robô, representada pelo sensor laser a bordo, a qual é de aproximadamente $7kg$. Portanto, para minimizar tais oscilações e, por conseguinte, melhorar a eficiência nas tarefas de seguimento, torna-se necessário agregar uma compensação da dinâmica do robô seguidor. Um procedimento adequado para fazer isto é descrito na Subseção 3.6.1, como possível extensão deste trabalho.

Vale ressaltar que a estratégia de controle descentralizado proposta neste capítulo pode ser expandida para uma formação de múltiplos robôs, utilizando duas abordagens distintas, a saber: estrutura líder-seguidores e estrutura líderes-seguidores. Tais abordagens são descritas na Subseção 3.6.2, assim como seus resultados de simulação.

3.6 Trabalhos Futuros

3.6.1 Consideração sobre a Dinâmica do Robô Uniciclo Aplicada à Formação

Os controladores projetados neste trabalho somente contemplam a cinemática do robô móvel, cujo modelo foi apresentado na Seção 1.1. Entretanto, esta seção abordará, embora de forma superficial, o modelo dinâmico de um robô móvel tipo uniciclo, e apresentará uma proposta para a compensação da dinâmica deste robô. Tal compensação pode ser agregada ao controlador descentralizado de formação implementado no robô seguidor, a fim de compensar a carga física inerente a ele e aquela que lhe é adicionada pela montagem do sensor laser. O objetivo é delinear uma possível extensão do trabalho aqui apresentado.

Em [30] é apresentado o modelo dinâmico de um robô móvel tipo unicycle, dado por

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{\nu} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \nu \cos \psi - a\omega \sin \psi \\ \nu \sin \psi + a\omega \cos \psi \\ \omega \\ \frac{\theta_3}{\theta_1}\omega^2 - \frac{\theta_4}{\theta_1}\nu \\ -\frac{\theta_5}{\theta_2}\nu\omega - \frac{\theta_6}{\theta_2}\omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{\theta_1} & 0 \\ 0 & \frac{1}{\theta_2} \end{bmatrix} \begin{bmatrix} \nu_{ref} \\ \omega_{ref} \end{bmatrix} + \begin{bmatrix} \delta_x \\ \delta_y \\ 0 \\ \delta_\nu \\ \delta_\omega \end{bmatrix}, \quad (3.37)$$

onde a especifica a que distância o ponto a ser controlado se encontra do eixo virtual que une as rodas. Tal distância deve ser obrigatoriamente diferente de zero. Em adição, $\theta = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ \theta_5 \ \theta_6]^T$ é o vetor de parâmetros do modelo, os quais podem ser identificados, e $\delta = [\delta_x \ \delta_y \ 0 \ \delta_\nu \ \delta_\omega]^T$ é um vetor de incertezas associadas ao modelo do robô móvel. Os sinais de controle ν_{ref} e ω_{ref} representam, respectivamente, as velocidades linear e angular definidas pelo controlador cinemático, enquanto ν e ω são os sinais de controle efetivamente desenvolvidos pelo robô móvel, cujos valores são obtidos de sua odometria durante a navegação.

Conforme apresentado em [46], o modelo da Equação 3.37 pode ser dividido em uma parte cinemática e uma parte dinâmica, como ilustrado na Figura 33. Por fim, são projetados controladores baseados em cada um destes modelos.

Desconsiderando o vetor de incertezas δ , a parte cinemática do modelo é dada por

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos \psi & -a \sin \psi \\ \sin \psi & a \cos \psi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \nu \\ \omega \end{bmatrix}, \quad (3.38)$$

cujas saídas representam as coordenadas do ponto de interesse a ser controlado, dados por $\mathbf{H} = [x \ y]^T$ (desprezando a orientação do mesmo). Por sua vez, a parte dinâmica é dada por

$$\begin{bmatrix} \dot{\nu} \\ \dot{\omega} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{1}{\theta_1} & 0 \\ 0 & \frac{1}{\theta_2} \end{bmatrix}}_{g(\theta)} \begin{bmatrix} \nu_{ref} \\ \omega_{ref} \end{bmatrix} + \begin{bmatrix} \frac{\theta_3}{\theta_1}\omega^2 - \frac{\theta_4}{\theta_1}\nu \\ -\frac{\theta_5}{\theta_2}\nu\omega - \frac{\theta_6}{\theta_2}\omega \end{bmatrix}, \quad (3.39)$$

cujas representação é semelhante àquela apresentada na Equação 3.19. Logo, propõe-se uma lei de controle dada pela Equação 3.21, que aqui pode ser escrita como

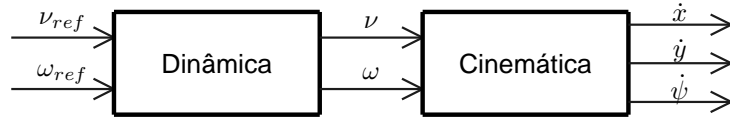


Figura 33: Diagrama de blocos do modelo dinâmico completo do robô móvel tipo uniciclo.

$$\begin{bmatrix} \nu_{ref} \\ \omega_{ref} \end{bmatrix} = \begin{bmatrix} \theta_1 & 0 \\ 0 & \theta_2 \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & -\omega^2 & \nu & 0 & 0 \\ 0 & 0 & 0 & 0 & \nu\omega & \omega \end{bmatrix} \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5 & \theta_6 \end{bmatrix}^T, \quad (3.40)$$

onde $\eta = [\eta_1 \ \eta_2]^T$ segue o conceito da Equação 3.22. Logo, tal vetor é dado por

$$\begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} \dot{\nu}_d \\ \dot{\omega}_d \end{bmatrix} + \begin{bmatrix} k_\nu & 0 \\ 0 & k_\omega \end{bmatrix} \begin{bmatrix} \tilde{\nu} \\ \tilde{\omega} \end{bmatrix}, \quad (3.41)$$

onde $k_\nu > 0, k_\omega > 0$ e os erros de velocidades linear e angular são $\tilde{\nu} = \nu_d - \nu$ e $\tilde{\omega} = \omega_d - \omega$. Em malha fechada, tem-se

$$\dot{\tilde{\mathbf{x}}} + \mathbf{K}\tilde{\mathbf{x}} = \mathbf{0},$$

ou seja,

$$\begin{bmatrix} \dot{\nu}_d \\ \dot{\omega}_d \end{bmatrix} - \begin{bmatrix} \dot{\nu} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} k_\nu & 0 \\ 0 & k_\omega \end{bmatrix} \begin{bmatrix} \tilde{\nu} \\ \tilde{\omega} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad (3.42)$$

cuja prova de estabilidade segue desenvolvimento semelhante àquele demonstrado na Subseção 3.2.2, e a conclusão é que $[\tilde{\nu} \ \tilde{\omega}]^T \in L_\infty^n \cap L_2^n$, ou seja, $[\tilde{\nu} \ \tilde{\omega}]^T \rightarrow [0 \ 0]^T$ para $t \rightarrow \infty$ [46].

O sistema descrito pela Equação 3.39 supõe que o vetor de parâmetros θ do robô é exato. Entretanto, em um controle de formação, os parâmetros dos robôs móveis do grupo podem sofrer alterações em virtude da tarefa que esteja sendo executada, como, por exemplo, o transporte de cargas. Neste caso, a massa do sistema varia nos instantes em que a carga a ser transportada é inserida ou retirada do sistema. Logo, torna-se essencial adaptar os parâmetros do sistema durante a execução da tarefa. Em [46] é apresentada a adaptação dos parâmetros que inclui a modificação *sigma*, bem como sua análise de estabilidade.

Finalmente, utilizando os sinais de controle corrigidos pelo controlador dinâmico, obtém-se a compensação da dinâmica do robô móvel. O diagrama de blocos da Figura 34 ilustra a estrutura de controle assim obtida. Vale enfatizar que o controlador cinemático

mostrado em tal figura é o controlador descentralizado de formação descrito na Seção 3.2, no caso do robô seguidor.

3.6.2 Extensão para Formações com Múltiplos Robôs

Esta subseção delinea como seria feita a extensão do controlador descentralizado de formação proposto na Seção 3.2 para aplicações utilizando mais de dois robôs. São tratadas aqui duas abordagens. A primeira refere-se a uma estrutura líder-seguidores, onde há somente um líder e vários seguidores que buscam o padrão montado sobre este robô líder. A estratégia de busca é aquela apresentada na Seção 3.3, a qual supõe que todos os seguidores dispõem de um sensor laser de varredura a bordo. A segunda abordagem considera uma estrutura líderes-seguidores, na qual todos os robôs que compõem a formação têm um padrão montado sobre si. Um dos robôs é eleito líder incondicional, o qual irá guiar a formação para, por exemplo, atingir um alvo. Já os demais robôs devem estabelecer a estrutura linear que lhes é designada, buscando o líder mais próximo.

3.6.2.1 Estrutura Líder-Seguidores

Quando a tarefa a ser executada exige que os robôs móveis estabeleçam uma estrutura geométrica predefinida, é essencial que os seguidores se posicionem relativamente a um guia, o qual foi eleito líder.

A seguir é apresentado um exemplo simulado (Figura 35), no qual o robô líder tem o padrão a ser identificado pelos três robôs seguidores, os quais são equipados com sensores laser de varredura (a razão de fazer somente simulação é que não há disponibilidade do número necessário de robôs e de sensores laser). A simulação consiste no controle

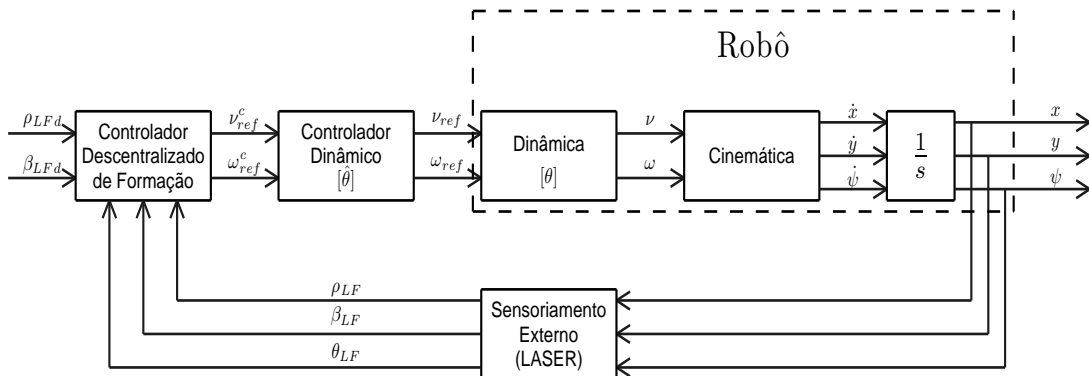


Figura 34: Estrutura de controle com compensação da dinâmica do robô móvel.

descentralizado dos robôs para que estes componham uma estrutura em forma de losango. É essencial, neste tipo de simulação, que todos os robôs seguidores possam *visualizar* o padrão montado no robô líder, para, assim, se posicionarem em relação a ele.

A tarefa a ser realizada pelo robô líder é atingir um alvo localizado nas coordenadas $(4000mm, 3000)$ partindo das coordenadas $(0mm, 0mm)$, com orientação inicial igual a zero. Para que esta tarefa seja cumprida, o controlador discutido na Subseção 1.3.1 foi implementado no mesmo. Os seguidores, nomeados F_1 , F_2 e F_3 , encontram-se inicialmente nas posições $(-750mm, 250mm)$, $(-500mm, -1000mm)$ e $(-1500mm, -250mm)$, com as respectivas orientações, $\psi_{F_1} = -15^\circ$, $\psi_{F_2} = 45^\circ$ e $\psi_{F_3} = 15^\circ$. A fim de se obter uma formação em losango, foram estabelecidos os valores desejados para as variáveis de formação, a saber: para o primeiro seguidor se definiu $\rho_{LFd} = 700mm$ e $\beta_{LFd} = 60^\circ$, para o segundo, $\rho_{LFd} = 700mm$ e $\beta_{LFd} = 120^\circ$, e para o terceiro $\rho_{LFd} = 1200mm$ e $\beta_{LFd} = 90^\circ$.

Como a estratégia de busca do líder e de controle de formação não contemplam a evasão de obstáculos, pode ocorrer, porventura, choques entre os entes do grupo durante a composição da formação. Isto se entende como colisão intra-formação, que não foi verificada na simulação realizada em virtude das condições iniciais e desejadas estipuladas.

Os resultados obtidos, apresentados aqui na Figura 36, demonstram a possibilidade de extensão da estrutura líder-seguidor para uma formação contendo mais de dois robôs. Isto é possível, pois cada par líder-seguidor pode ser tratado como uma estrutura linear distinta, o que resultará na forma global desejada, neste caso um losango. A Figura 36(a) ilustra o caminho traçado pelo pelotão para atingir a formação desejada. Pode-se visu-

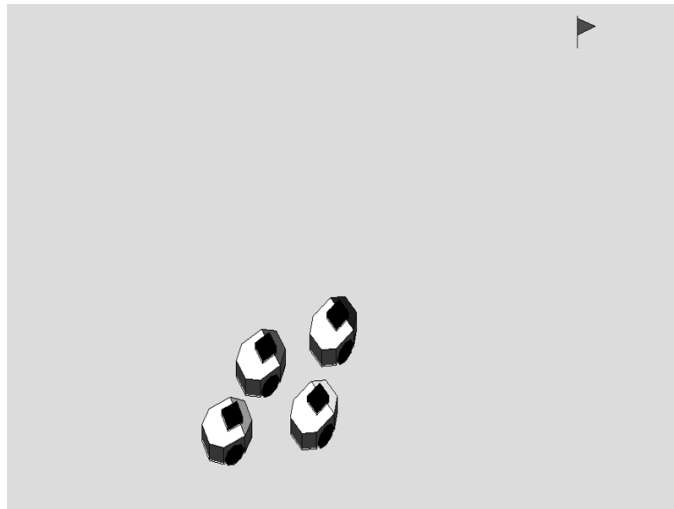


Figura 35: Formação Multi-robôs em losango.

alizer, facilmente, que os robôs se organizaram com êxito em uma estrutura em losango, cumprindo, portanto, o objetivo da tarefa. Vale destacar, também, que o robô líder atingiu o alvo que lhe foi proposto. Os robôs da formação foram plotados em intervalos de 10s.

A evolução das variáveis de formação dos robôs seguidores pode ser vista na Figura 36(b). Constatou-se que os robôs F_1 e F_2 tenderam assintoticamente ao mesmo valor desejado de ρ_{LF} , apresentando, entretanto, uma convergência estável a ângulos de formação, β_{LF} , distintos. Por outro lado, é possível visualizar facilmente, por parte do robô F_3 , a tendência assintótica das variáveis de formação controladas aos valores desejados.

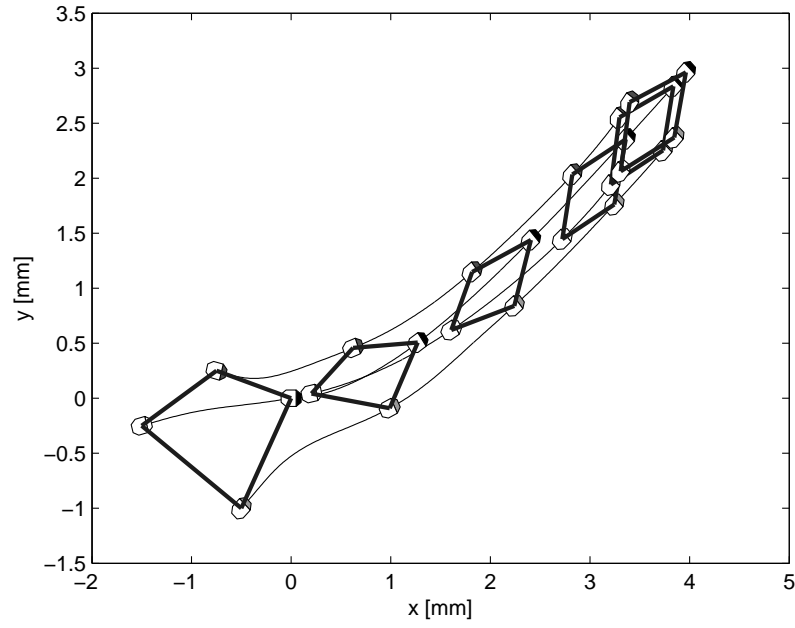
A Figura 36(c) apresenta os sinais de controle enviados aos robôs da formação. É possível verificar que, uma vez em estado estacionário, as velocidades linear e angular enviadas aos seguidores praticamente se sobrepõem àquelas enviadas ao líder da formação, ocasionando, portanto, um movimento suave de todo o pelotão. Esta mesma figura ainda ilustra a velocidade linear ν_L do líder estimada por todos seguidores. Apesar de visíveis oscilações, verifica-se um seguimento da tendência de ν_L , crucial para a tarefa de seguimento.

Este tipo de estratégia de formação multi-robôs pode ser usada em tarefas de transporte de cargas, onde é essencial o conhecimento prévio da postura dos robôs na formação, e este posicionamento deve ser assegurado durante toda a navegação. Afinal, qualquer deformação poderia comprometer o cumprimento da tarefa, pois ocasionaria a queda da carga.

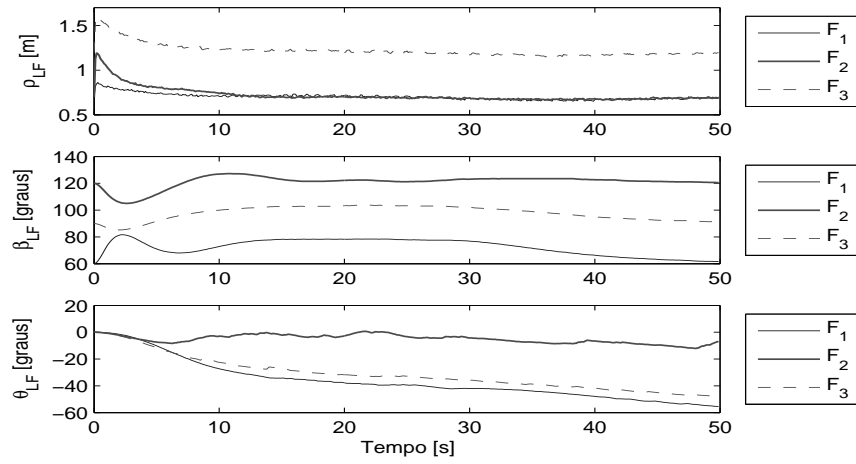
3.6.2.2 Estrutura Líderes-Seguidores

Diferentemente da estrutura líder-seguidores, a estrutura aqui apresentada não exige que cada robô seguidor busque sua postura em relação a um líder único. Para isto, em todos os robôs (líder e seguidores) é montado o padrão já apresentado no decorrer deste trabalho, assim como um sensor que seja capaz de extrair as variáveis de formação através da identificação do padrão montado no “líder” mais próximo. Para a simulação apresentada a seguir, considerou-se que todos os robôs possuem sensores laser de varredura.

Na estrutura líderes-seguidores, é primordial que a um dos robôs seja atribuído o papel de líder incondicional, e a ele compete cumprir a tarefa. Esta, por sua vez, pode ser atingir um alvo no plano de navegação, por exemplo. Aos demais robôs compete a tarefa de constituir e manter a formação.



(a) Caminho percorrido pela formação.



(b) Evolução das variáveis de formação.

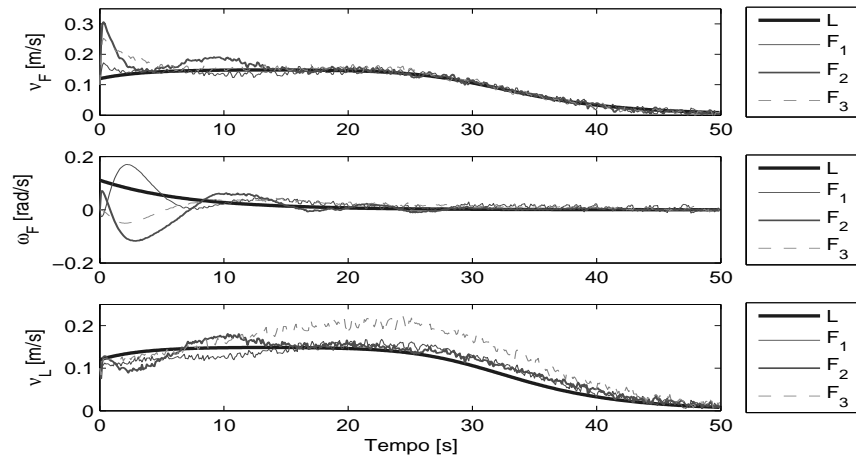
(c) Sinais de controle e velocidade linear v_L estimada do robô líder.

Figura 36: Simulação: estrutura líder-seguidores com formação em losango.

Um exemplo de estrutura líderes-seguidores pode ser visto na Figura 37, onde cada robô situado imediatamente à frente de um outro robô passa a ser o líder deste último, até que se defina todos os pares de formação líder-seguidor. Vale destacar que o robô posicionado à frente de todos os demais é o líder incondicional.

Para validar a estrutura aqui proposta foi realizada uma simulação, cujos resultados são apresentados na Figura 38. Nesta simulação, o líder incondicional, munido do controlador de posição, deve atingir a posição $(4000mm, 3000mm)$ no plano \mathbf{XY} de navegação, partindo das coordenadas iniciais $(0mm, 0mm)$, com orientação $\psi_L = 0$. Os demais robôs executam o controlador descentralizado de formação, todos com valores desejados $\rho_{LFd} = 700mm$ e $\beta_{LFd} = 90^\circ$.

A Figura 38(a) apresenta o caminho percorrido pelos robôs, a fim de formar a estrutura líder-seguidor que lhes foi definida. Os robôs da formação foram plotados em intervalos de $15s$. Os caminhos traçados podem causar a impressão de que os seguidores do líder incondicional deveriam percorrer o mesmo caminho seguido por ele. Entretanto, como existem vários líderes, cada seguidor busca o seu e se forma perante este. A evolução de todas as variáveis de formação, mostrada na Figura 38(b), destaca o fato de que para todos os seguidores as variáveis de formação convergem assintoticamente para os valores desejados.

Os sinais de controle enviados durante a tarefa de posicionamento podem ser vistos na Figura 38(c). É possível visualizar que, nas proximidades do alvo, os sinais de controle tendem a zero, indicando, portanto, o cumprimento assintoticamente estável da tarefa.

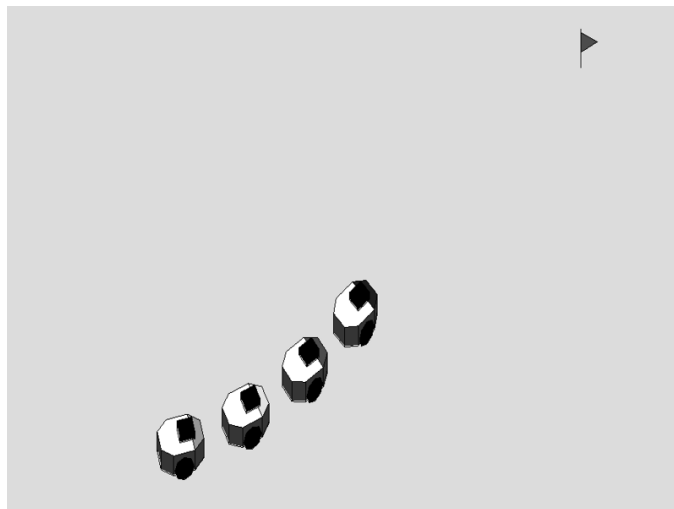
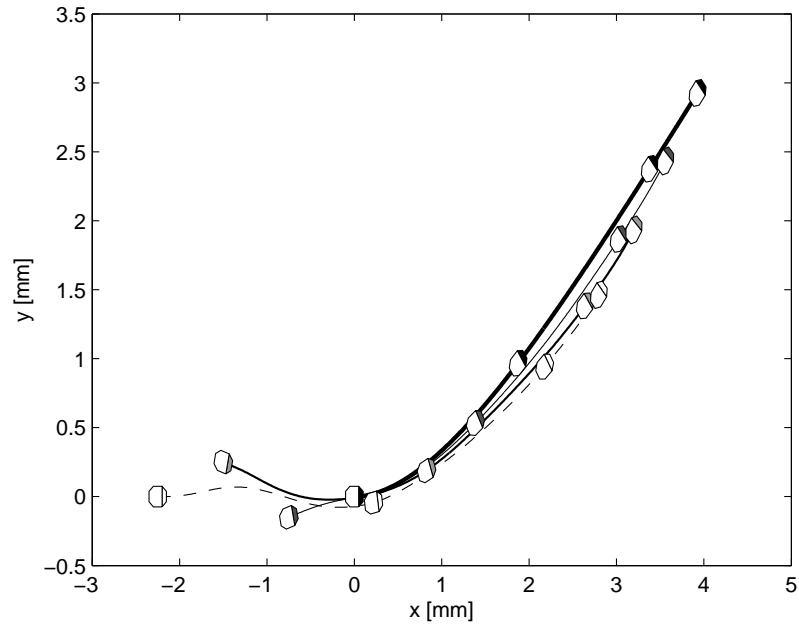


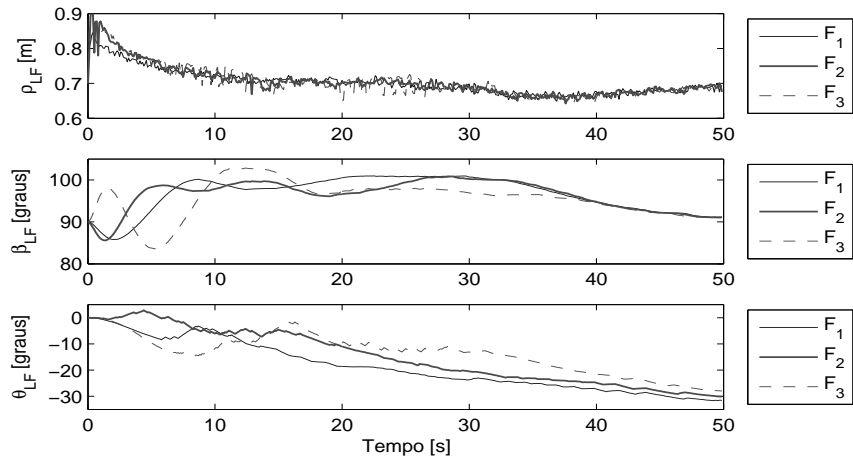
Figura 37: Formação Multi-robôs em linha.

de posicionamento, e, conseqüentemente, da tarefa de seguimento. Observe-se, também, que ao atingir o alvo os seguidores permanecem em formação. Adicionalmente, na Figura 38(c), é apresentada a velocidade linear ν_L estimada pelos seguidores. Constata-se claramente que a estimação seguida da filtragem por Kalman é eficiente no acompanhamento da evolução de ν_L .

A navegação multi-robôs utilizando a estrutura líderes-seguidores pode ser usada, por exemplo, para mapeamento de ambientes, pois uma possível deformação não comprometeria a execução da tarefa. Outro exemplo de aplicação seria a busca e desarmamento de minas terrestres, pois se um ente da formação se perder durante a busca, os demais componentes poderiam perfeitamente prosseguir com a execução da tarefa.



(a) Caminho percorrido pela formação.



(b) Evolução das variáveis de formação.

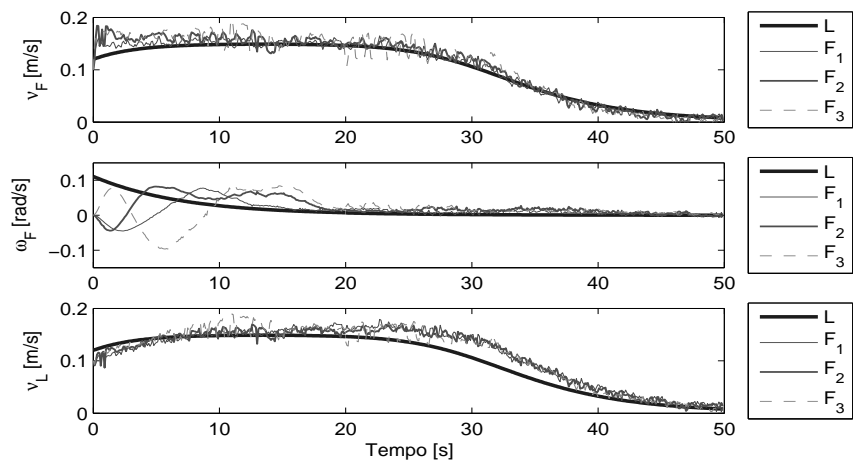
(c) Sinais de controle e velocidade linear ν_L estimada do robô líder.

Figura 38: Simulação: estrutura líderes-seguidores em formação linear.

4 *Desvio de Obstáculo Aplicado à Formação*

*“Em momentos de crise,
só a imaginação é mais importante
que o conhecimento.”*

(Albert Einstein)

Durante a navegação, para cumprir uma tarefa proposta, a formação pode se deparar com situações nas quais é necessário evitar obstáculos antes de alcançar o objetivo. As abordagens já apresentadas na literatura usualmente consideram a formação como um corpo totalmente rígido ou totalmente flexível, classificação esta que está diretamente relacionada à tarefa a ser realizada. Neste contexto, uma formação rígida se assemelha a uma estrutura virtual, pois os robôs são submetidos individualmente a severas restrições de movimento determinadas pela lei de controle de formação que previne qualquer deformação estrutural. Um exemplo de tarefa que requer uma formação estritamente rígida é o transporte de cargas. Em [47] é proposto e experimentalmente validado um sistema de controle para navegação de uma formação rígida apta a evadir obstáculos. Foi proposto, neste sentido, um perímetro mínimo, que envolve todos os robôs do grupo, o qual permite a execução de trajetórias seguras e adequadas à tarefa de desvio de obstáculos.

No caso de uma formação flexível, existe a possibilidade dos robôs desfazerem a formação para evitar um obstáculo e se reorganizarem após ultrapassá-lo. Dentre as tarefas já mencionadas, podemos citar busca e resgate e inspeção de áreas como tarefas adequadas para uma formação flexível. Em [48] é proposta uma estratégia de evasão de obstáculos para uma formação flexível inspirada nos fenômenos hidrodinâmicos. A formação, inicialmente rígida, modifica sua forma nas proximidades de um obstáculo, de forma similar ao movimento de fluidos, que se alongam e se aproximam na presença de obstruções.

Quanto ao sensoriamento utilizado e à estratégia de evasão de obstáculos utilizada,

alguns trabalhos da literatura podem ser mencionados. Em [28, 29, 40] campos potenciais são utilizados para a realização da tarefa de desvio de obstáculo com uma formação descentralizada, através da definição de zonas de repulsão (obstáculos) e atração (pontos de destino). Em [26] e [49], sistemas Fuzzy baseados nas informações de um scanner laser e um sistema omnidirecional, respectivamente, foram implementados em simulações para obter uma formação líder-seguidor, evadir obstáculos e evitar colisão inter-formação. Tais sistemas se baseiam nas medidas de proximidade provenientes de seus sensores e de um conjunto de regras a elas associadas. Em [41], desvio de obstáculos combinando campos potenciais e seguimento de paredes foi testado experimentalmente em uma formação flexível centralizada. Em [50] é apresentado um sistema comutado entre aprendizado por reforço e algoritmos genéticos, o qual é aplicado individualmente aos robôs de uma formação flexível para evitar obstáculos ou, então, para restabelecer a estrutura geométrica de navegação.

Este capítulo discute duas abordagens para desvio de obstáculos de uma formação descentralizada de robôs móveis. Primeiramente, é apresentada uma estratégia de evasão aplicada a uma formação rígida. Tendo em vista o sistema líder-seguidor apresentado na Seção 3.1, nesta estratégia a estrutura linear não pode ser deformada, o que quer dizer que, em estado estacionário, ρ_{LF} e β_{LF} devem atingir e permanecer nos valores desejados, independentemente se durante a navegação existir ou não obstruções físicas na rota seguida por qualquer um dos robôs que compõem a formação. Posteriormente, assume-se que a formação descentralizada é do tipo semi-rígida, ou seja, há uma restrição rígida em relação à distância entre os robôs, ρ_{LF} , enquanto o ângulo da formação, β_{LF} , apresenta uma flexibilidade controlável. Ao agregar tais características, a formação semi-rígida torna-se apta a evitar obstáculos de forma tão eficiente quanto a formação rígida. Entretanto, ela é capaz de transpor obstáculos cujas configurações tornam inviáveis a aplicação de estratégias com restrições rígidas de movimento (como um corredor estreito, por exemplo).

O capítulo em questão está organizado como segue: a Seção 4.1 apresenta a estratégia de desvio de obstáculos para a formação rígida. Resultados de simulações e experimentais são apresentados, a fim de validar a estratégia proposta. A Seção 4.2 trata da estrutura semi-rígida de formação descentralizada, bem como a estratégia de desvio de obstáculos aplicada a este tipo de formação. Ao final desta seção são apresentados resultados de simulação e experimentais comparativos com o métodos apresentado na Seção 4.1. Por fim, a Seção 4.3 apresenta conclusões a respeito das abordagens de desvio de obstáculos aplicadas à formação descentralizada de robôs móveis tratadas neste capítulo, assim como algumas sugestões para trabalhos futuros.

4.1 Desvio de Obstáculos para uma Formação Rígida

Nesta seção é proposta uma estratégia de desvio de obstáculo para uma formação rígida. O objetivo é utilizar o algoritmo de desvio tangencial modificado implementado somente no robô líder da formação, garantindo a não colisão do robô seguidor com o obstáculo a ser evitado. Vale ressaltar que o sistema sensorial à bordo do robô seguidor é destinado à busca, reconhecimento e identificação do robô líder, ou seja, à sua localização no ambiente.

A estrutura líder-seguidor daqui em diante será denominada robô virtual, o qual irá evitar os obstáculos. Para esta estratégia de desvio de obstáculo com formação rígida é imprescindível conhecer previamente a distância entre os robôs que constituem a estrutura linear (ρ_{LF}), conjuntamente com a distância mínima de reação (d_{obs}) especificada no algoritmo de desvio tangencial. Com estas informações é possível determinar a distância de reação a um obstáculo para o robô virtual, dada por $d_F = \rho_{LF} + d_{obs}$. A Figura 39 ilustra tal estratégia.

4.1.1 Simulação e Discussão

Esta subseção apresenta uma simulação do método proposto, a fim de demonstrar sua viabilidade. A tarefa a ser desempenhada é atingir um ponto alvo localizado na posição (10m,0m) do espaço de trabalho do robô líder. Este robô deve cumprir tal tarefa evitando obstáculos presentes no meio. A Figura 40(a) ilustra o ambiente 3D de simulação, destacando os robôs líder e seguidor, bem como um terceiro robô (mais à direita) que

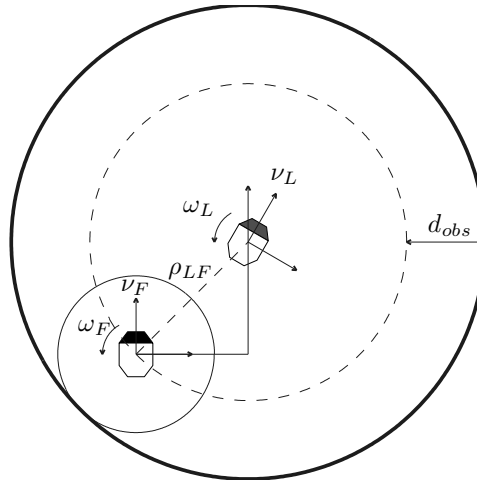


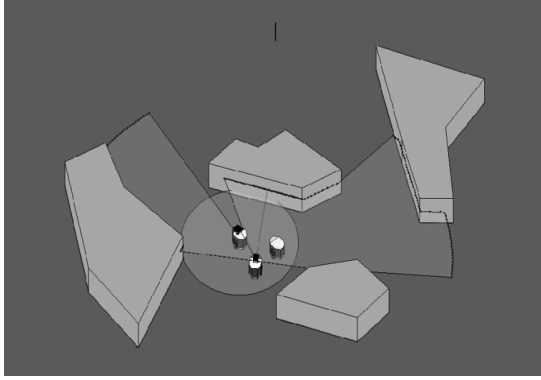
Figura 39: Abordagem de desvio tangencial aplicada à formação rígida.

representa a posição ideal do robô seguidor, o qual não apresenta restrições holonômicas. A Figura 40(b) apresenta a reconstrução do ambiente de navegação utilizando os dados provenientes do sensor laser do robô seguidor durante a tarefa de seguimento. Em adição, são apresentados os caminhos percorridos pelos robôs da formação, os quais foram plotados em intervalos de 15s. A estrutura da formação exigida nesta simulação foi $\rho_{LF} = 1m$ e $\beta_{LF} = 120^\circ$.

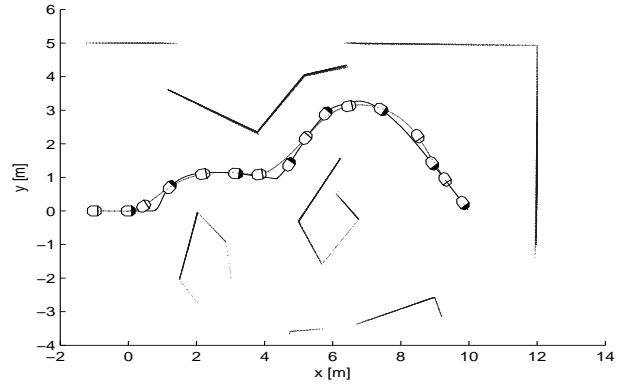
A Figura 40(c) apresenta a evolução das variáveis de formação durante o cumprimento da tarefa de posicionamento do robô líder e seguimento pelo robô seguidor. Verifica-se que a distância ρ_{LF} entre os robôs da formação tende ao valor desejado, apresentando oscilações máximas de $\pm 50mm$, uma vez estabelecida a formação. Entretanto, o ângulo da formação β_{LF} sofre alterações significativas quando a formação se encontra na situação de desvio de obstáculos. Porém, observa-se que uma vez vencido o obstáculo, $\beta_{LF} \rightarrow \beta_{LFd}$, fato que ocorre na etapa de conclusão da tarefa. A variável de formação θ_{LF} , a qual não é controlada diretamente, apresenta um bom comportamento, ou seja, oscila indicando os erros de orientação entre os robôs e tende a um valor limitado, quando a formação se encontra em estado estacionário e na ausência de obstáculos.

A Figura 40(d) apresenta os sinais de controle (ν_F e ω_F) enviados ao robô seguidor para que este possa realizar a tarefa de seguimento, bem como a velocidade linear estimada do robô líder (ν_L), essencial no cálculo de tais sinais de controle. Observa-se, ao final da tarefa, quando o robô líder se aproxima do alvo, que ambas as velocidades tendem a zero, indicando que o robô seguidor mantém a formação mesmo quando o robô líder se encontra estático (tarefa de estacionamento). A velocidade linear do robô líder foi estimada e suavizada utilizando Filtro de Kalman, em decorrência do fato de não existir comunicação entre os robôs, conforme mencionado anteriormente. Pode-se perceber que esta segue a tendência de ν_L com a presença de ruídos inerentes ao processo de estimação, o qual é realizado discretizando a Equação 3.3.

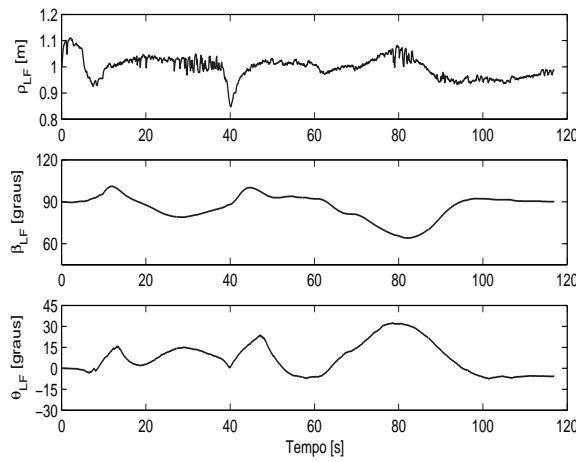
Conforme mencionado e explicado, a distância ρ_{LF} apresenta variações aceitáveis durante toda a tarefa. Entretanto, o ângulo de formação β_{LF} se deforma consideravelmente. Neste sentido, seria interessante fazer um controle de formação semi-rígida, onde ρ_{LF} permanecesse inalterado e β_{LF} pudesse oscilar durante a tarefa de evasão de obstáculos. As mudanças no ângulo de formação fariam com que o sistema se assemelhasse a um sistema mola-amortecedor, o qual oscilaria em função da distância ao obstáculo, visando realizar manobras de evasão mais seguras. Neste contexto, o robô seguidor também deveria realizar a tarefa de sensoramento do ambiente de navegação na busca de obstáculos.



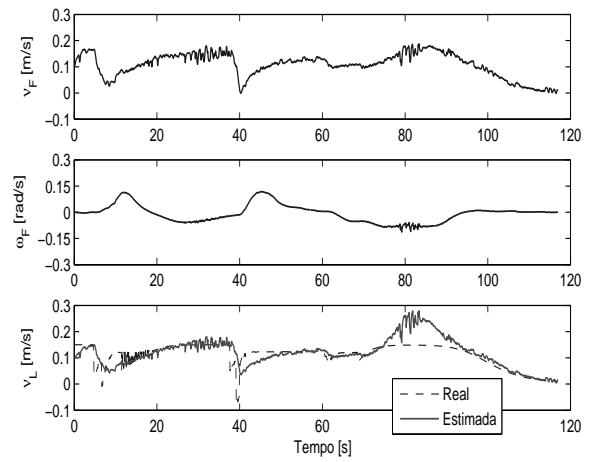
(a) Ambiente de Navegação 3D com obstáculos.



(b) Percurso dos robôs e sensoriamento laser realizado pelo robô seguidor durante a tarefa.



(c) Evolução das variáveis de formação.



(d) Sinais de controle do robô seguidor e velocidade linear estimada do robô líder.

Figura 40: Simulação 1 para a formação Rígida.

4.1.2 Experimento e Discussão

Nesta subseção é apresentado um experimento realizado com uma formação líder-seguidor em uma configuração linear rígida numa tarefa de evasão de obstáculos. Aqui, as variáveis de formação que definem a distância entre os robôs, ρ_{LF} , e o ângulo da formação, β_{LF} , não podem sofrer modificações durante a execução da tarefa. A formação está definida como $\rho_{LFd} = 500mm$ e $\beta_{LFd} = 90^\circ$. No ambiente de navegação, o obstáculo consiste de uma parede inicialmente localizada à direita da formação, e, posteriormente, de paredes laterais, as quais definem um corredor.

A Figura 41(a) apresenta o caminho percorrido por ambos os robôs a fim de cumprir a tarefa desejada, a qual neste caso é levar o robô líder da posição inicial $(0mm, 0mm)$ até o alvo localizado nas coordenadas $(3000mm, -3000mm)$, desviando de obstáculos, enquanto o robô seguidor estabelece a formação desejada. As coordenadas iniciais do

robô seguidor são $(-700mm, 0mm)$.

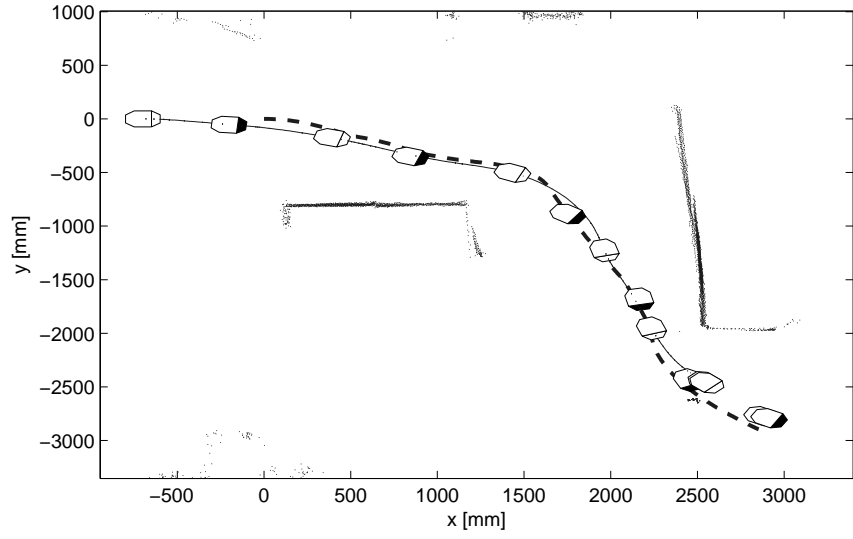
É importante mencionar que o algoritmo de desvio de obstáculo aqui utilizado é o *Desvio Tangencial Modificado* apresentado na Seção 2.2, entretanto, como não há disponibilidade de dois sensores de varredura laser para serem usados em ambos os robôs da formação, a estratégia foi adaptada para implementação utilizando um *array* de 8 sensores de ultra-som, disponíveis na parte frontal do robô Pioneer 2DX utilizado como robô líder.

A linha tracejada na Figura 41(a) representa o caminho executado pelo robô líder para cumprir a tarefa que lhe foi imposta, enquanto a linha contínua representa o caminho percorrido pelo robô seguidor. As ilustrações dos robôs líder (em preto) e seguidor (em branco) foram realizadas em intervalos de 10s, sendo que a postura do robô líder é plotada segundo as variáveis de formação, ou seja, representa a postura como é *vista* pelo robô seguidor.

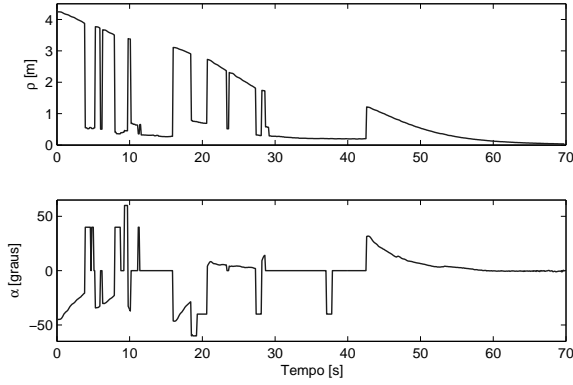
Os erros de controle do robô líder, que consistem no erro de posição, ρ , e de orientação, α , tendem a zero, conforme podem ser vistos na Figura 41(b), assim como estava previsto na análise de estabilidade do controlador de posição apresentada na Seção 1.3.1. Os sinais de controle enviados e efetivamente desenvolvidos pelo robô líder são apresentados na Figura 41(c) e estes são os responsáveis pelo cumprimento da tarefa. É possível verificar que os mesmos tendem a zero nas proximidades do alvo, devido à dependência direta das variáveis de controle que, conforme visto na Figura 41(b), também tendem a zero. Na Figura 41(b), são percebidas variações abruptas no erro de posição, isto se deve ao fato de que, na presença de um obstáculo, a distância a este é assumida como erro de distância.

Quanto ao robô seguidor, a evolução das variáveis de formação é apresentada na Figura 41(d). É possível verificar oscilações sobre o valor desejado da distância entre os robôs, ρ_{LF} , menores que $\pm 50mm$, quando em estado estacionário, e menores que $\pm 100mm$, quando em situações onde o robô líder está desviando de obstáculo e, nestes casos, o robô seguidor, ainda assim, deve manter a formação. Em relação ao ângulo de formação, β_{LF} , as variações são em torno de 20° , o que na prática comprometeria a navegação em uma formação que deve ser rígida. Entretanto, em estado estacionário, na ausência de obstáculos, as variáveis de formação controláveis tendem aos valores desejados e o erro de orientação entre os robôs, θ_{LF} , tende a um valor constante.

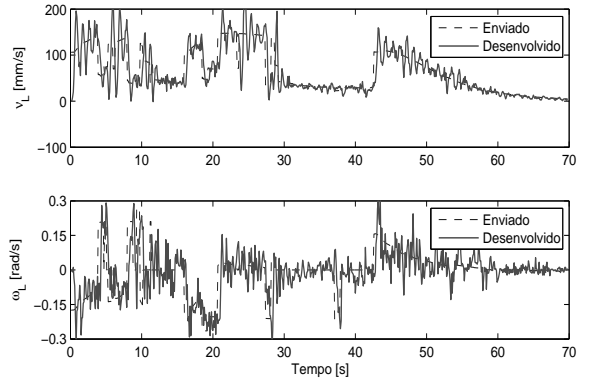
Os sinais de controle enviados e desenvolvidos pelo robô seguidor são apresentados na Figura 41(e), assim como a velocidade linear ν_L estimada do robô líder. A convergência assintótica dos sinais de controle a zero se deve à dependência direta dos erros das variáveis de formação, os quais tendem a zero.



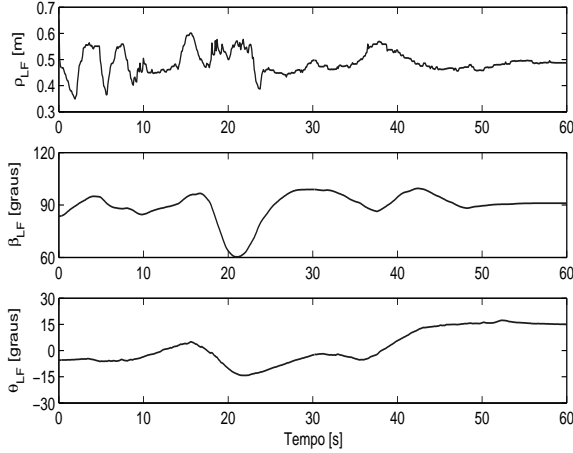
(a) Caminho circular percorrido pelos robôs em formação.



(b) Variáveis de erro do controlador de posição do líder.



(c) Sinais de controle enviados e efetivamente desenvolvidos pelo robô líder.



(d) Evolução das variáveis de formação.

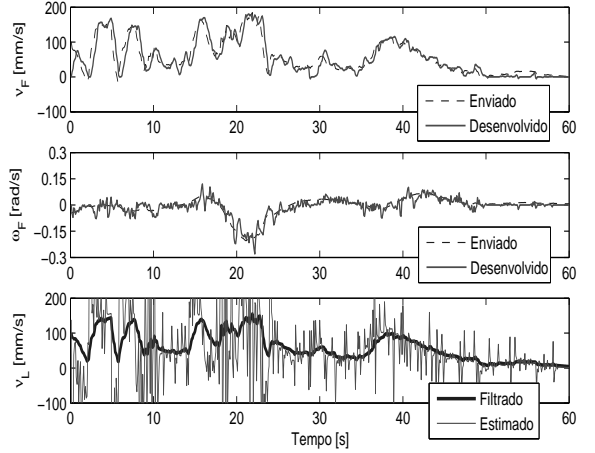
(e) Sinais de controle enviados e efetivamente desenvolvidos pelo robô seguidor, e velocidade linear v_L estimada do robô líder.

Figura 41: Experimento formação rígida com desvio de obstáculos.

Vale comentar que, para esta configuração de obstáculo, uma formação líder-seguidor configurada para $\beta_{LFd} = 120^\circ$, não seria possível de ser executada, pois, possivelmente, ocorreria um colisão com as paredes laterais. Experimentos para comprovar esta afirmação não foram realizados visando preservar a integridade dos robôs. A formação semi-rígida enunciada na Seção 4.2 é apresentada para sanar esta debilidade.

4.2 Desvio de Obstáculos para uma Formação Semi-Rígida

A proposta de uma estratégia de sensoriamento mútuo do ambiente pelos robôs líder e seguidor auxiliaria a tarefa de evasão de obstáculos em ambientes estreitos, pois a formação rígida exige um robô virtual de grandes dimensões. Isto, por sua vez, restringe ou impossibilita a evasão da formação em algumas configurações de obstáculos. Neste sentido, esta seção apresenta uma estratégia de desvio de obstáculos para a formação líder-seguidor em uma configuração semi-rígida. Isto significa que a distância entre os robôs líder e seguidor dever ser mantida fixa, enquanto o ângulo da formação pode ser alterado, visando permitir as manobras de evasão de obstáculos.

Diferentemente do caso apresentado na Seção 4.1, não existe a figura do robô virtual, pois os objetivos desta estratégia são tornar desnecessário o conhecimento do valor de ρ_{LF} por parte do robô líder e reduzir sua circunferência de reação aos obstáculos. Para atingir tais objetivos, adotou-se o sensoriamento do ambiente por ambos os robôs, a fim de detectar possíveis obstáculos, os quais podem comprometer a tarefa de seguimento e, conseqüentemente, a estrutura da formação. Neste caso, a dimensão da região de segurança do robô líder passa a ser igual a d_{obs} , ao passo que o raio da região de observação do robô seguidor é dado por $d_{max} = \rho_{LF} + d_{obs}$. A Figura 42 ilustra tal configuração, especificando as regiões referentes a cada robô da formação.

O algoritmo de desvio tangencial modificado apresentado na Seção 2.2 é aplicado como tal ao robô líder. Por outro lado, o robô seguidor utiliza sua informação sensorial para alterar o valor desejado do ângulo de formação β_{LF} . Esta alteração se assemelha ao controle por impedância mecânica apresentado em [47] para tarefas de desvio de obstáculos em tempo real, cujo fundamento é associar o movimento do robô móvel com as forças de interação entre sua posição atual e seu entorno. Tendo em conta que a idéia de desvio de obstáculo é evitá-los sem qualquer contato físico, a interação com o meio é definida por

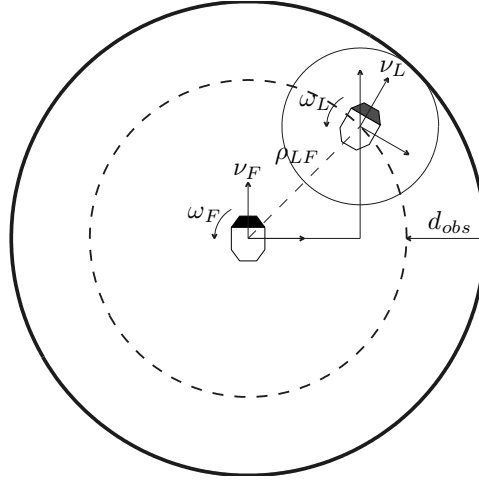


Figura 42: Regiões de reação de cada robô em uma formação semi-rígida.

uma força fictícia F_f dada por

$$F_f = a - b(d - d_{min})^n \quad (4.1)$$

onde d é a distância entre o robô e os obstáculos, d_{min} é a distância mínima sem contato do robô com o obstáculo, n é um expoente positivo e a e b são constantes positivas que devem respeitar

$$a - b(d_{max} - d_{min})^n = 0, \quad (4.2)$$

onde d_{max} é o limite da zona de repulsão do obstáculo.

A relação dinâmica entre as forças fictícias e os ângulos de rotação para evitar um obstáculo é dada por

$$e = Z^{-1}F_f,$$

onde Z representa a impedância mecânica do meio e é dada por

$$Z = Is^2 + Bs + K,$$

onde I , B e K são constantes positivas que representam, respectivamente, a inércia, o amortecimento e a constante elástica da impedância mecânica. A Figura 43 ilustra a estratégia de desvio de obstáculos por controle baseado em impedância.

Adaptando a estratégia de controle por impedância ao desvio de obstáculos da formação semi-rígida, as forças fictícias calculadas segundo a Equação 4.1 são divididas entre a observação à direita e à esquerda do robô seguidor, tendo em vista a localização do padrão montado no robô líder, conforme mostrado na Figura 44.

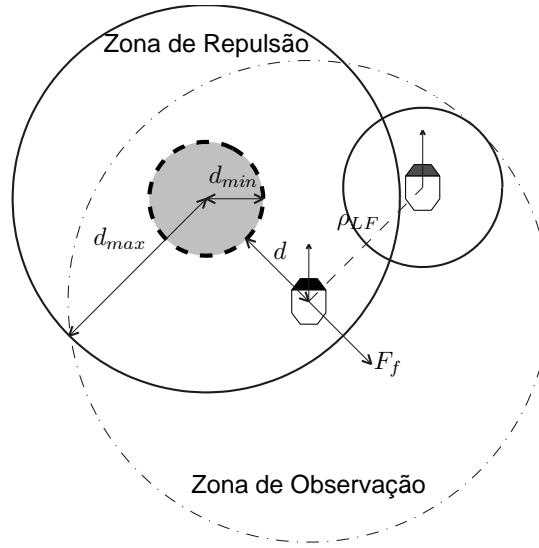


Figura 43: Estratégia de desvio de obstáculos baseado em impedância mecânica.

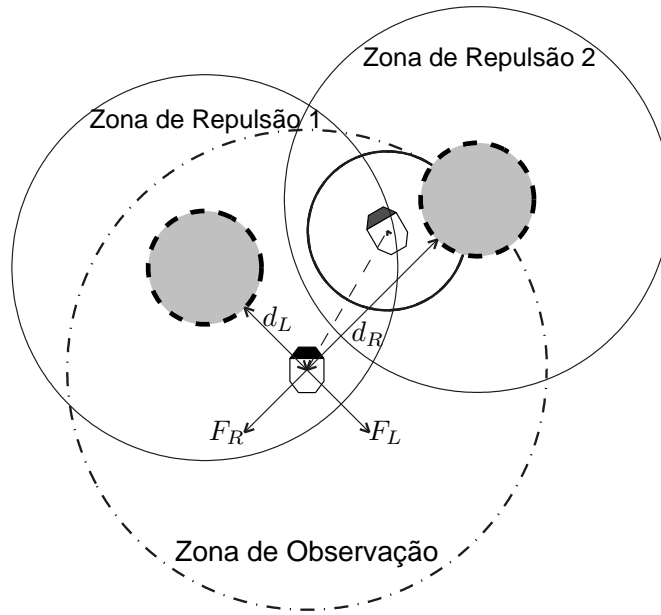


Figura 44: Forças fictícias à direita e à esquerda do robô seguidor.

As variáveis d_R e d_L , mostradas na Figura 44, representam a medida de distância mais próxima à direita e à esquerda do robô seguidor. A estas medidas estão associadas às forças fictícias, F_R e F_L , respectivamente. Para normalização das forças foram adotados os valores: $a = 1,00$, $b = 0,51$ e $n = 2$, para $d_{min} = 0,3m$ e $d_{max} = 1,7m$, conforme restrição imposta pela Equação 4.2. Finalmente, assume-se que o valor desejado do ângulo da formação líder-seguidor na presença de obstáculos será dado por

$$\beta_{LFd} = 90^\circ - \beta_R + \beta_L, \quad (4.3)$$

onde $\beta_R = K^{-1}F_R$ e $\beta_L = K^{-1}F_L$, uma vez considerado que a impedância mecânica do meio é simplesmente elástica, com $K = \frac{1}{30^\circ}$.

A Equação 4.3 expressa a essência da formação semi-rígida, que é modificar o ângulo de formação desejado e manter fixa a distância entre os robôs. Além disso, esta estratégia sana as dificuldades encontradas pela formação rígida ao navegar por ambientes estreitos na presença de múltiplos obstáculos ou ao se transladar em corredores (dependendo especificamente do ângulo da formação desejado). Afinal, quando o robô seguidor detecta um obstáculo à sua direita, a força de interação gerada neste mesmo lado resultará em um ângulo β_R , proporcional à força, que alterará β_{LFd} , a fim de repelir o robô seguidor para a região do ambiente onde não há obstruções. Para obstáculos à esquerda do robô seguidor, a situação ocorre ao revés, tratando, neste caso, do ângulo β_L . Entretanto, se o robô seguidor se deparar com a situação em que receba a influência de forças de interação tanto à direita quanto à esquerda, β_{LFd} se modifica de modo que o seguidor, no pior dos casos, tenha que seguir o líder estando exatamente atrás deste, ou seja, $\beta_{LFd} = 90^\circ$. Esta abordagem é adotada em função da confiança depositada no robô líder, o qual tem implementado o algoritmo de desvio tangencial modificado (apresentado no Capítulo 2 e em [51]) que lhe garante uma navegação segura. Logo, o caminho já percorrido por este robô pode ser perfeitamente seguido por outro, o que de fato ocorre quando $\beta_{LFd} = 90^\circ$. Vale enfatizar que para esta estratégia ρ_{LFd} não é alterado.

4.2.1 Simulação e Discussão

Esta subseção é dedicada à apresentação dos resultados simulados utilizando a técnica de desvio tangencial para uma formação semi-rígida. Para efeito de comparação com o método apresentado na Seção 4.1, a primeira simulação foi realizada em um ambiente com características similares às aquelas ilustradas na Figura 40(a). A Figura 45(a) apresenta o ambiente de simulação em uma perspectiva tridimensional, enquanto a Figura 45(b) ilustra os caminhos percorridos pelos robôs líder e seguidor durante o cumprimento da tarefa. O objetivo desta simulação é levar o robô líder da posição $(0mm, 0mm)$ até a posição $(10000mm, 0mm)$ utilizando o algoritmo de desvio tangencial modificado para evitar obstáculos. Por sua vez, o robô seguidor deve preservar a formação definida por $\rho_{LF} = 1m$ e um ângulo de formação β_{LF} variável, conforme estabelecido na estratégia da formação semi-rígida ($\beta_{LF} = 90^\circ$, inicialmente). Os robôs, como aparecem em tal figura, foram plotados em intervalos de 15s.

Comparando a Figura 45(b) com a Figura 40(b), é possível verificar que a trajetória

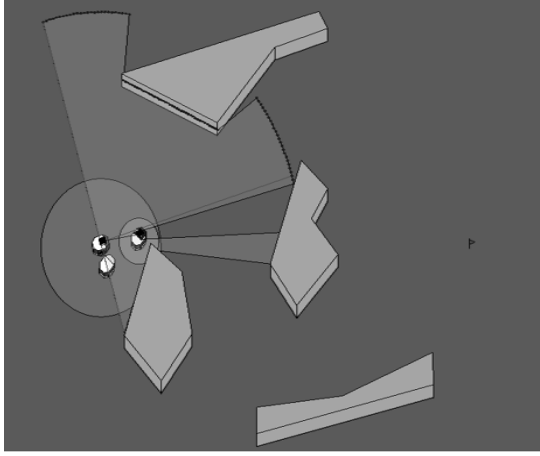
do robô líder se aproxima mais dos obstáculo em função da redução das dimensões do robô virtual mencionado na Seção 4.1, que neste caso não existe. Em virtude desta nova abordagem a formação é capaz de evitar obstáculos que estão dispostos mais próximos entre si. Tratando agora do robô seguidor, verifica-se que este realiza movimentos seguros, ou seja, não se aproxima excessivamente dos obstáculos durante as manobras de evasão.

A Figura 45(c) apresenta a evolução das variáveis de formação durante o cumprimento da tarefa. Verifica-se uma variação significativa na variável β_{LF} , a fim de suavizar os movimentos do seguidor durante o desvio de obstáculos. Quando em regime estacionário, porém, $\beta_{LF} \rightarrow \beta_{LFd} = 90^\circ$. A distância entre os robôs, ρ_{LF} , apresenta uma oscilação média de $\pm 50mm$ sobre o valor desejado, quando o sistema está em estado estacionário. Porém, ela sofre alterações severas quando o robô líder detecta um obstáculo, pois a estimativa da velocidade do robô líder, devido ao seguimento da tendência, não é alterada abruptamente (ver Figura 45(d)) como ocorre com a velocidade linear deste robô ao reagir a um obstáculo. O erro de orientação entre os robôs da formação θ_{LF} apresenta um bom comportamento, sendo limitado quando o sistema se encontra em estado estacionário, além de não apresentar descontinuidades durante a execução da tarefa.

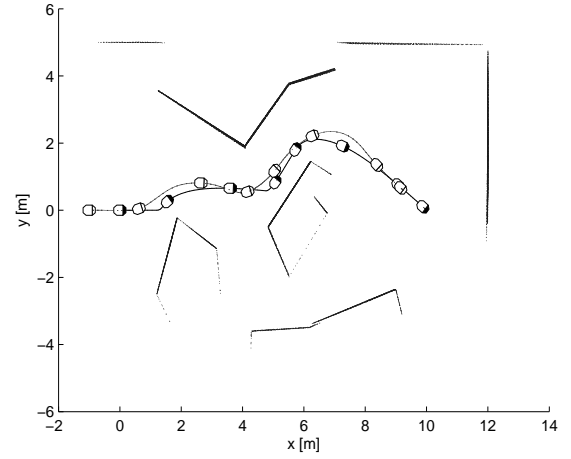
A evolução dos sinais de controle enviados ao robô seguidor pode ser observada na Figura 45(d). Conforme previsto, uma vez que o robô líder se aproxima do alvo proposto, a tarefa de seguimento passa a ser considerada como uma tarefa de estacionamento, e assim os sinais de controle tendem a zero, à medida que a velocidade linear do líder se reduz a zero e as variáveis de formação tendem aos valores desejados.

A segunda simulação trata da situação na qual os obstáculos estão muito próximos, o que torna inviável a utilização de uma formação de característica rígida. A Figura 46(a) ilustra de forma tridimensional o ambiente de simulação e a Figura 46(b) destaca as trajetórias percorridas por ambos os robôs da formação, sendo o ambiente reconstruído a partir das informações de distância provenientes do sensor de varredura laser do robô seguidor. É importante destacar que a dificuldade de evasão não é exclusiva do controlador de formação a bordo do seguidor, mas também do algoritmo de desvio tangencial implementado no robô líder. No entanto, quando o líder evita os obstáculos navegando entre eles, o seguidor executa uma trajetória com um ângulo de formação igual a 90° , ressaltando o fato de se ter um caminho seguro, uma vez que este já foi seguido por outro robô (neste caso, o líder).

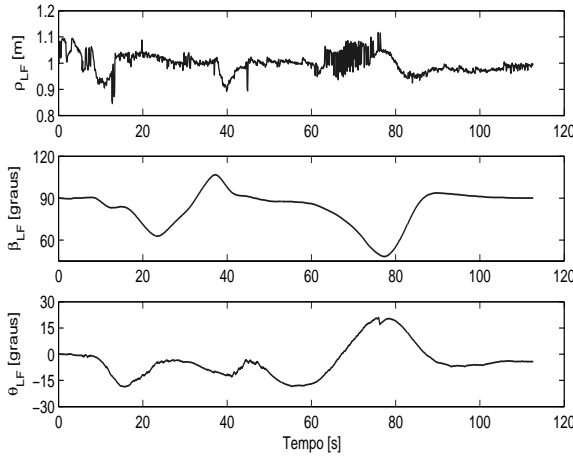
As Figuras 46(c) e 46(d) ilustram, respectivamente, a evolução das variáveis de formação e os sinais de controle enviados ao robô seguidor. Destaca-se que, independente



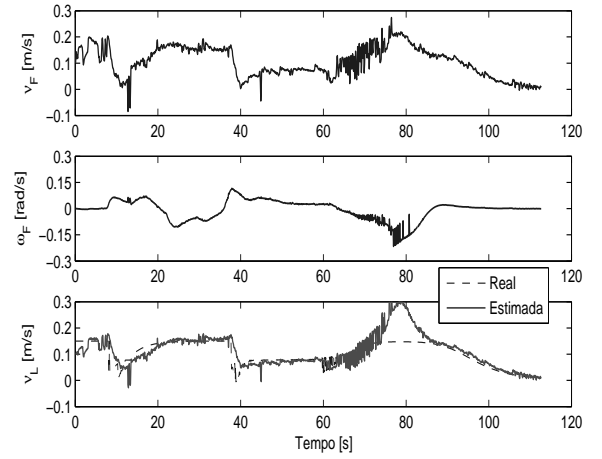
(a) Ambiente de Navegação 3D com obstáculos.



(b) Percurso dos robôs e sensoriamento laser realizado pelo robô seguidor durante a execução da tarefa.



(c) Evolução das variáveis de formação.



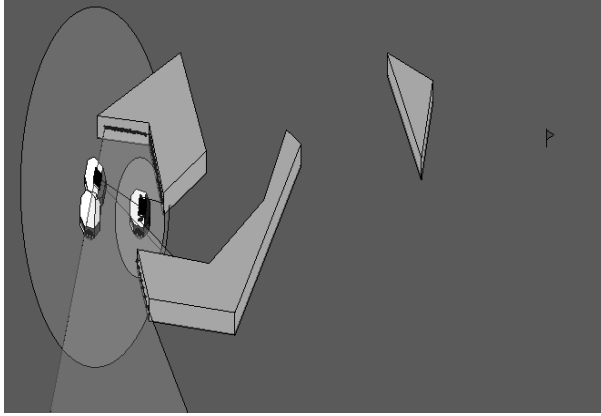
(d) Sinais de controle do robô seguidor e velocidade linear estimada do robô líder.

Figura 45: Simulação 1 para a formação Semi-rígida.

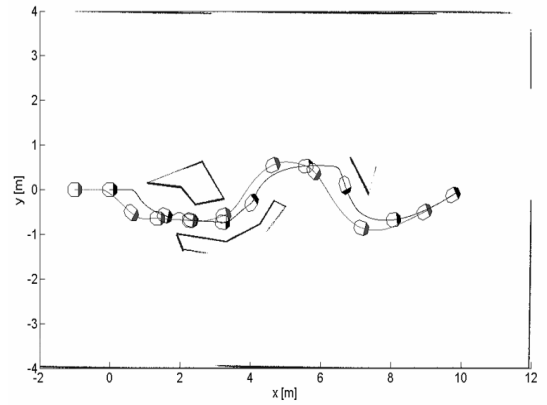
da configuração do ambiente, as variáveis de formação controladas tendem aos valores desejados ($\rho_{LFd} = 1m$ e $\beta_{LFd} = 90^\circ$) na ausência de perturbações (obstáculos), durante a busca do alvo pelo robô líder. E, de forma semelhante à apresentada na simulação anterior, os sinais de controle do seguidor tendem a zero, à medida que o robô líder se aproxima do objetivo proposto.

4.2.2 Experimento e Discussão

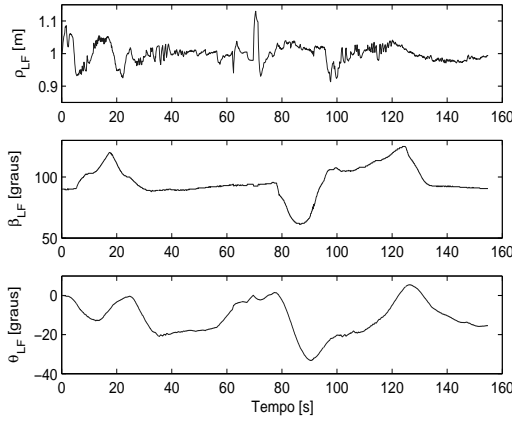
O experimento apresentado nesta subseção é idêntico aquele apresentado na Subseção 4.1.2, porém neste caso é utilizada a abordagem de desvio de obstáculo aplicada a uma formação semi-rígida.



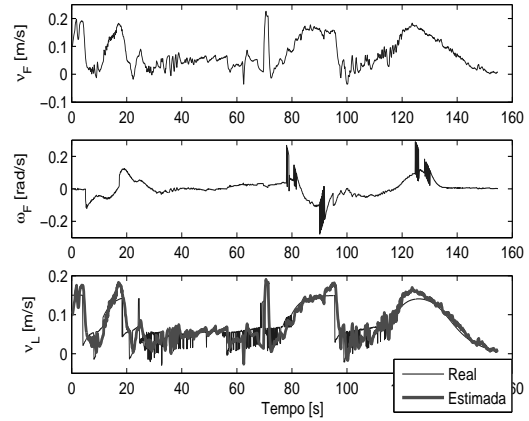
(a) Ambiente de Navegação 3D com obstáculos.



(b) Percurso dos robôs e sensoriamento laser realizado pelo robô seguidor durante a execução da tarefa.



(c) Evolução das variáveis de formação.



(d) Ações de controle do robô seguidor e velocidade linear estimada do robô líder.

Figura 46: Simulação 2 para a formação Semi-rígida.

A Figura 47(a) apresenta os caminhos percorridos por ambos os robôs da formação. Os robôs são plotados em intervalos de 10s, sendo que a postura do robô líder (em preto) é mostrada segundo a *visão* corrente do robô seguidor (em branco). Em comparação ao caminho apresentado na Figura 41(a), o caminho traçado pelo robô seguidor, utilizando a abordagem semi-rígida, é mais próximo dos obstáculos, porém esta aproximação é realizada de forma segura, devido às modificações nos ângulos de formação desejado.

A tarefa de posicionamento prevista pelo robô líder é executada com êxito, conforme pode vista na Figura 47(a) e comprovada pela tendência assintótica a zero dos erros de posição, ρ , e de orientação, α . Os sinais de controle enviados e efetivamente desenvolvidos pelo robô líder para o cumprimento da tarefa pode ser visto na Figura 47(c), os mesmos tendem a zero devido a dependência direta dos erros de controle, conforme previsto pela Equação 1.8.

A evolução das variáveis de formação, durante a tarefa de seguimento e desvio de obstáculos, pode ser visualizada na Figura 47(d). Constata-se que as oscilações na distância entre os robôs, cujo valor desejado é $\rho_{LFd} = 500mm$, não são superiores $\pm 100mm$, uma vez que a formação já está constituída, o que ocorre após os primeiros 5s de experimento. Na ausência de obstáculos, em estado estacionário, as oscilações são inferiores a $\pm 50mm$, como ocorre após 60s do experimento. Quanto ao ângulo de formação β_{LF} , este sofre variações de $\pm 30^\circ$, conforme previsto na estratégia de desvio de obstáculos para uma formação semi-rígida.

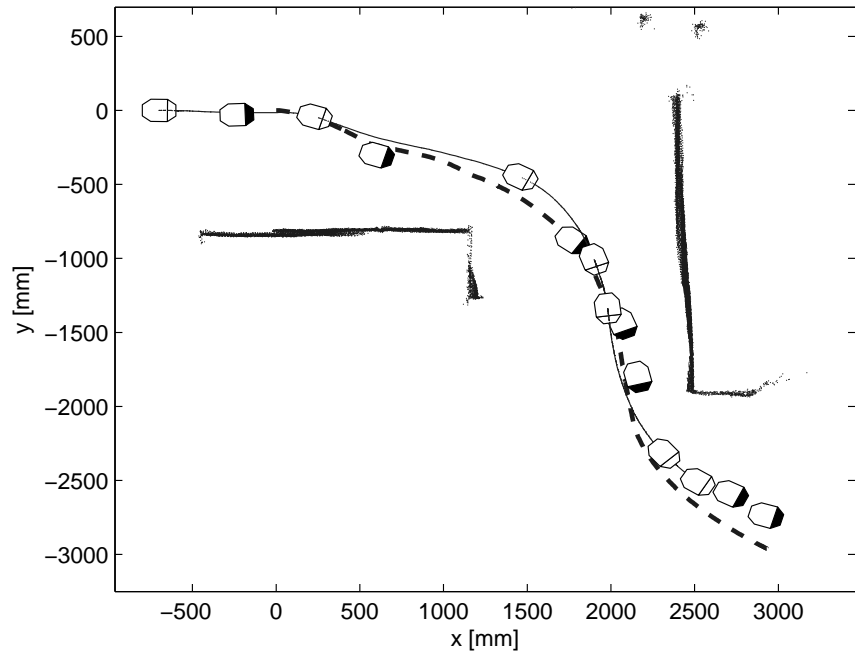
Os sinais de controle enviados e efetivamente desenvolvidos e a velocidade linear ν_L estimada do robô líder são apresentados na Figura 47(e). É possível verificar que o robô líder após cumprir a tarefa de posicionamento, o robô seguidor, por sua vez, deixar de executar uma tarefa de seguimento e passa a executar uma tarefa de estacionamento, que neste caso consiste em permanecer com os sinais de controle próximos de zero como ocorre após 58s de experimento, ressaltando que ν_L estimada também apresenta valores próximos de zero.

4.3 Conclusões e Trabalhos Futuros

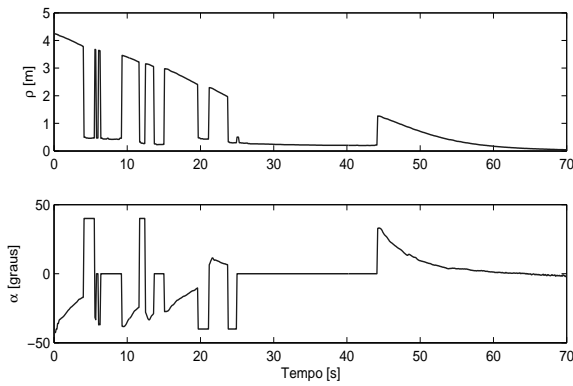
Através do controlador de formação proposto é possível garantir a convergência assintótica das variáveis de formação aos valores desejados, resultando na formação linear líder-seguidor, durante a busca de um ponto alvo. Na presença de obstáculos, a utilização da estratégia de desvio tangencial modificado aplicada a uma formação rígida se mostrou aplicável e eficiente quando há obstáculos no ambiente de navegação. Diversas situações foram simuladas, a fim de comprovar a viabilidade do método proposto.

Para múltiplos obstáculos próximos entre si, a estratégia de formação semi-rígida se mostrou eficiente, devido à ausência do robô virtual de grandes dimensões essencial na formação rígida. Tal estratégia conta com a observação do ambiente por ambos os robôs da formação, e possibilita evitar obstáculos intransponíveis pela formação rígida. Resultados de simulação foram também apresentados a fim de validar a proposta e ilustrar as situações na qual o robô seguidor realiza trajetórias com um ângulo de formação de 90° , aproveitando os caminhos seguros já percorridos pelo robô líder.

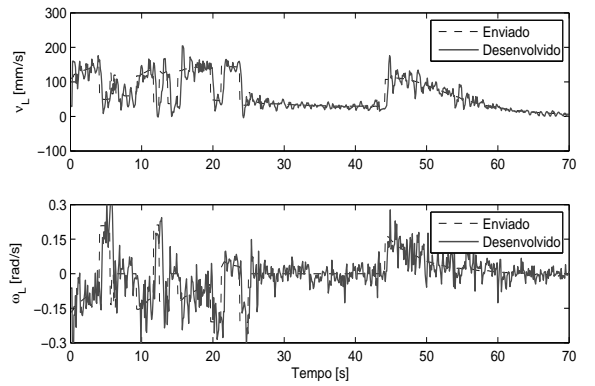
Para trabalhos futuros, pretende-se projetar e implementar a estratégia de desvio de obstáculos com uma formação flexível em atividades de inspeção, onde cada robô do grupo é dotado de estratégias de desvio tangencial e de reconstrução da formação em função do estado atual do ambiente de navegação.



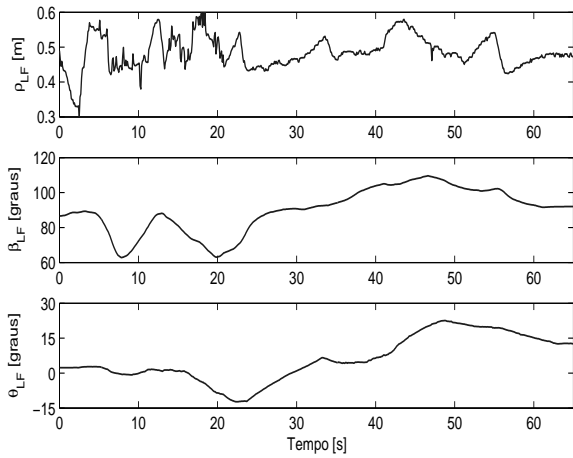
(a) Caminho circular percorrido pelos robôs em formação.



(b) Variáveis de erro do controlador de posição do líder.



(c) Sinais de controle enviados e efetivamente desenvolvidos pelo robô líder.



(d) Evolução das variáveis de formação.

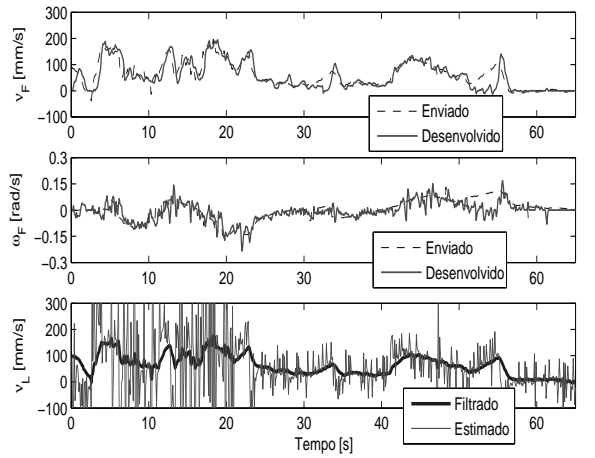
(e) Sinais de controle enviados e efetivamente desenvolvidos pelo robô seguidor, e velocidade linear v_L estimada do robô líder.

Figura 47: Experimento formação semi-rígida com desvio de obstáculos.

Referências

- [1] FERREIRA, A. *Desvio Tangencial de Obstáculos para um Robô Móvel Navegando em Ambientes Semi-Estruturados*. Dissertação (Mestrado) — Universidade Federal do Espírito Santo, Vitória - ES, Brasil, Junho 2004.
- [2] SECCHI, H. A. *Control de Vehículos Autoguiados con Realimentación Sensorial*. Dissertação (Mestrado) — Universidad Nacional de San Juan, San Juan, Argentina, Julho 1998.
- [3] MARCHI, J. *Navegação de Robôs Móveis Autônomos: Estudo e Implementação de Abordagens*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis, SC, Fevereiro 2001.
- [4] JONES, J. L.; FLYNN, A. M.; SEIGER, B. A. *Mobile Robots: Inspiration to implementation*. Second edition. Natick, Massachusetts: A K Peters, 1999.
- [5] PEREIRA, F. G. *Navegação e Desvio de Obstáculos Usando um Robô Móvel Dotado de Sensor de Varredura Laser*. Dissertação (Mestrado) — Universidade Federal do Espírito Santo, Vitória - Brasil, 2006.
- [6] SECCHI, H. A.; CARELLI, R.; MUT, V. A. An experience on stable control of mobile robots. In: *Latin American Applied Research*. Buenos Aires, Argentina: LAAR, 2003. v. 33, n. 4, p. 379–385.
- [7] TOIBERO, J. M. et al. Switching controllers for navigation with obstacles in unknown environments. In: *Anales de la IV Jornadas Argentinas de Robótica (JAR'06)*. Córdoba, Argentina: JAR, 2006.
- [8] VIDYASAGAR, M. *Nonlinear System Analysis*. 2nd. ed. New Jersey: Prentice Hall, 1993.
- [9] KUC, R.; BARSHAN, B. Navigating vehicles through an unstructured environment with sonar. In: *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*. Scottsdale, AZ: IEEE, 1989. v. 3, p. 1422–1426.
- [10] ELFES, A. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, RA-3, n. 3, p. 249–265, June 1987.
- [11] KHATIB, O. Real time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, v. 5, n. 1, p. 90–98, 1986.
- [12] BORENSTEIN, J.; KOREN, Y. Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 19, n. 5, p. 1179–1187, 1989.

- [13] BORENSTEIN, J.; KOREN, Y. The vector field histogram - fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, v. 7, n. 3, p. 278–288, 1991.
- [14] MINGUEZ, J.; MONTANO, L. Nearness diagram (nd) navigation: collision avoidance in troublesome scenarios. *IEEE Transactions on Robotics and Automation*, v. 20, n. 1, p. 45–59, 2004.
- [15] MINGUEZ, J.; MONTANO, L. Sensor-based robot motion generation in unknown, dynamic and troublesome scenarios. *Robotics and Autonomous Systems*, v. 52, n. 4, p. 290–311, 2004.
- [16] LAMIRAUX, F.; BONNAFOUS, D.; LEFEBVRE, O. Reactive path deformation for nonholonomic mobile robots. *IEEE Transactions on Robotics*, v. 20, n. 6, p. 967–977, 2004.
- [17] BELKHOUS, S. et al. A novel approach for mobile robot navigation with dynamic obstacles avoidance. *Journal of Intelligent and Robotic Systems*, v. 44, n. 3, p. 187–201, 2005.
- [18] BELKHOUCHE, F.; BELKHOUCHE, B. A method for robot navigation toward a moving goal with unknown maneuvers. *Robotica*, v. 23, n. 6, p. 709–720, 2005.
- [19] YAGI, Y. et al. Reactive visual navigation based on omnidirectional sensing - path following and collision avoidance. *Journal of Intelligent and Robotic Systems*, v. 31, n. 4, p. 379–395, 2001.
- [20] WILLMS, A. R.; YANG, S. X. An efficient dynamic system for real-time robot-path planning. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, v. 36, n. 4, p. 755–766, 2006.
- [21] YANG, S. X.; MENG, M. Neural network approaches to dynamic collision-free trajectory generation. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, v. 31, n. 3, p. 302–318, 2001.
- [22] QU, Z.; WANG, J.; PLAISTED, C. E. A new analytical solution to mobile robot trajectory generation in the presence of moving obstacles. *IEEE Transactions on Robotics*, v. 20, n. 6, p. 978–993, 2004.
- [23] KIM, K.-H.; CHO, H. S. An obstacle avoidance method for mobile robots based on fuzzy decision-making. *Robotica*, v. 24, n. 5, p. 567–578, 2006.
- [24] ER, M. J.; DENG, C. Obstacle avoidance of a mobile robot using hybrid learning approach. *IEEE Transactions on Industrial Electronics*, v. 52, n. 3, p. 898–905, 2005.
- [25] FERREIRA, A. et al. Avoiding obstacles in mobile robot navigation: Implementing the tangential escape approach. In: *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE'06)*. Montreal, Canadá: IEEE, 2006. v. 4, p. 2732–2737.
- [26] SISTO, M.; GU, D. A fuzzy leader-follower approach to formation control of multiple mobile robots. In: *Proceedings of the 2006 IEEE/RSJ, International Conference on Intelligent Robots and Systems (IROS'06)*. Beijin, China: IEEE, 2006. p. 2515–2520.

- [27] STILWELL, D.; BISHOP, B. A strategy for controlling autonomous robot platoons. In: *Proceedings of the 39th IEEE Conference on Decision and Control*. Sydney, Australia: IEEE, 2000. v. 4, p. 3483–3488.
- [28] WANG, J.; WU, X.-B.; XU, Z.-L. Decentralized formation control and obstacles avoidance based on potential field method. In: *Proceedings of the 50th International Conference on Machine Learning and Cybernetics*. Dalian: IEEE, 2006. p. 803–808.
- [29] LIANG, Y.; LEE, H.-H. Decentralized formation control and obstacle avoidance for multiple robots with nonholonomic constraints. In: *Proceedings of the 2006 American Control Conference (ACC'06)*. Minnesota, USA: IEEE, 2006.
- [30] CRUZ, C. de da; CARELLI, R. Dynamic modeling and centralized formation control of mobile robots. In: *Proceedings of the 32nd Annual Conference of the IEEE Industrial Electronics Society (IECON 2006)*. Paris, France: IEEE, 2006. p. 3880–3885.
- [31] CHEN, Y. Q.; WANG, Z. Formation control: a review and a new consideration. In: *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'05)*. Edmonton, Canada: IEEE, 2005. p. 3181–3186.
- [32] BRANDAO, A. S. et al. Controlling a mobile robot to keep a line formation using a laser scanner. In: *Anales de la XII Reunión de Trabajo en Procesamiento de la Información y Control (XII RPIC)*. Santa Cruz, Argentina: UNPA, 2007.
- [33] GAVA, C. et al. Nonlinear control techniques and omnidirectional vision for team formation on cooperative robotics. In: *Proceedings of the 2007 IEEE International Conference on Robotics and Automation (ICRA'07)*. Roma, Itália: IEEE, 2007. p. 2409–2414.
- [34] LAWTON, J. R. T.; BEARD, R. W.; YOUNG, B. J. A decentralized approach to formation maneuvers. *IEEE Transaction on Robotics and Automation*, v. 19, n. 6, p. 933–941, December 2003.
- [35] DONG, W.; GUO, Y.; FARRELL, J. Formation control of nonholonomic mobile robots. In: *Proceedings of the 2006 American Control Conference (ACC'06)*. Minnesota, USA: IEEE, 2006.
- [36] SHAO, J. et al. Leader-following formation control of multiple mobile robots. In: *Proceedings of the 2005 IEEE International Symposium on Intelligent Control (ISIC'05)*. Limassol, Cyprus: IEEE, 2005. p. 808–813.
- [37] OGREN, P.; LEONARD, N. Obstacle avoidance in formation. In: *Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA'03)*. Texas, USA: IEEE, 2003. v. 2, p. 2492–2497.
- [38] JIA, Q.; LI, G.; LU, J. Formation control and attitude cooperative control of multiple rigid body systems. In: *Proceedings of the 60th International Conference on Intelligent Systems Design and Applications (ISDA'06)*. Jinan, China: IEEE, 2006. v. 2, p. 82–86.
- [39] TAN, K.-H.; LEWIS, M. Virtual structures for high-precision cooperative mobile robotic control. In: *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'96)*. Osaka, Japan: IEEE, 1996. v. 1, p. 132–139.

- [40] DE-GENNARO, M.; JADBABAIE, A. Formation control for a cooperative multi-agent system using decentralized navigation functions. In: *Proceedings of the 2006 American Control Conference (ACC'06)*. Minnesota, USA: IEEE, 2006.
- [41] LIU, B.; ZHANG, R.; SHI, C. Formation control of multiple behavior-based robots. In: *Proceedings of the International Conference on Computational Intelligence and Security (CIS'06)*. Guangzhou, China: Springer, 2006. v. 1, p. 544–547.
- [42] WANG, Z.; TAN, D.; GOLDSMITH, P. Formation control of robotic vehicles. In: *Proceedings of the IEEE International Conference on Robotics and Biomimetics (RO-BIO'05)*. Shatin, China: IEEE, 2005. p. 57–60.
- [43] LUMELSKY, V. J.; HARINARAYAN, K. R. Decentralized motion planning for multiple mobile robots: The cocktail party model. *Autonomous Robots*, v. 4, p. 121–135, 1997.
- [44] ARRICHIELLO, F. *Coordination Control of Multiple Mobile Robots*. Tese (Doctor) — Università Degli Studi Di Cassino, November 2006.
- [45] BRANDAO, A. S. et al. Controle de robôs móveis para navegação em formação linear usando um sensor de varredura laser. In: *Anais do VIII Simpósio Brasileiro de Automação Inteligente (VIII SBAI)*. Florianópolis, SC: SBA, 2007.
- [46] MARTINS, F. N. et al. Controlador dinâmico adaptativo de seguimento de trajetória para robôs móveis. In: *Anais do VIII Simpósio Brasileiro de Automação Inteligente (VIII SBAI)*. Florianópolis, SC: SBA, 2007.
- [47] CRUZ, C. de la. *Control de Formación de Robots Móviles*. Tese (Doctorado) — Universidad Nacional de San Juan, San Juan, Argentina, December 2006.
- [48] SHAO, J.; WANG, L.; XIE, G. Flexible formation control for obstacle avoidance based on numerical flow field. In: *Proceedings of the 45th IEEE Conference on Decision and Control*. San Diego, CA, USA: IEEE, 2006. p. 5986–5991.
- [49] KOPFSTEDT, T. et al. Formation control for mobile robots in partially known environments using mixed integer programming and fuzzy systems. In: *Proceeding of the International Joint Conference SICE-ICASE*. Busan, Korea: Editorial Board of Control Theory & Applications, 2006. p. 1832–1837.
- [50] KOBAYASHI, F.; TOMITA, N.; KOJIMA, F. Re-formation of mobile robots using genetic algorithm and reinforcement learning. In: *Proceedings of the SICE 2003 Annual Conference*. Fukui, Japan: IEEE, 2003. v. 3, p. 2902–2907.
- [51] BRANDAO, A. S. et al. A simple approach to avoid obstacles in mobile robot navigation: The tangential escape. *IEEE/ASME Transactions on Mechatronics (manuscrito em análise)*, 2007.

ANEXO A – Filtro de Kalman

Quando se busca um sistema determinístico para um processo qualquer, primeiramente se constrói um modelo do sistema baseado nas leis físicas que regem sua dinâmica. Em seguida, é essencial estudar sua estrutura e sua resposta em função de entradas conhecidas. Por fim, quando necessário, projetar controladores para alterar as características do sistema, bem como para controlá-los. Os modelos determinísticos são definidos como sendo uma aproximação da realidade, pois, normalmente, não contemplam as perturbações e as imperfeições dos sensores.

Entretanto, a dúvida que resta é como modelar sistemas que levam em conta as incertezas associadas, bem como estimar de forma ótima os dados de interesse destes sistemas mal modelados e corrompidos por ruídos.

O método de Kalman, descrito por Rudolf E. Kalman, em 1958, com o intuito de estimar estados de um sistema estocástico, é apresentado neste anexo, a fim de compensar os processos aleatórios que circundam um sistema determinístico.

O filtro de Kalman é um algoritmo ótimo recursivo para processamento de dados. É dito ótimo pois minimiza um critério determinado (função custo), incorporando toda a informação do processo para então determinar a resposta filtrada. É dito recursivo porque não necessita manter os dados prévios, facilitando, assim, sua implementação em tempo real. Por fim, é um algoritmo de processamento de dados, pois se trata de um filtro. O objetivo do filtro de Kalman é estimar de forma ótima os estados de um processo estocástico, minimizando o erro médio quadrático da predição.

A.1 Representação do Sistema

Certas classes de sinais podem ser representadas por

$$Z_n = a_{n-1}Z_{n-1} + W_{n-1}, \text{ para } n = 1, 2, \dots, \quad (\text{A.1})$$

onde Z_0 é uma variável aleatória no tempo inicial, a_n é uma seqüência conhecida de constantes e W_n é uma seqüência de variáveis aleatórias de média zero descorrelacionadas e de variância $E[W_n^2]$, que pode depender do tempo. O processo (ou sistema) Z_n é geralmente não estacionário, e não está disponível diretamente. Por outro lado, é possível realizar observações deste processo, as quais são representadas por

$$X_n = Z_n + N_n, \text{ para } n = 1, 2, \dots, \quad (\text{A.2})$$

onde N_n é um ruído de observação de média zero, descorrelacionado e de variância $E[N_n^2]$, que pode depender do tempo. Os ruídos de observação, N_n , e do sistema, W_n , são descorrelacionados para todo instante n_1 e n_2 .

Com o objetivo de encontrar uma estimativa para Z_n , baseada nas informações *a priori* de X_0, X_1, \dots, X_{n-1} , utiliza-se um estimador linear que pode variar no tempo, dado por

$$Y_n = \sum_{j=1}^n h_j^{(n-1)} X_{n-j}, \text{ onde } h_j^{(n-1)} \text{ são os parâmetros do filtro ótimo.} \quad (\text{A.3})$$

Para que um filtro seja ótimo, este deve satisfazer a Condição de Ortogonalidade, que diz que o erro ε_n deve ser ortogonal a todas as observações X_n , ou seja,

$$E[\varepsilon_n X_l] = 0, \forall l, \quad (\text{A.4})$$

onde $\varepsilon_n = Z_n - Y_n$ é aqui chamado de erro de predição. Logo, substituindo a Equação A.3 em A.4 tem-se que

$$E[(Z_n - \sum_{j=1}^n h_j^{(n-1)} X_{n-j}) X_l] = 0, \forall l = 0, 1, \dots, n-1,$$

ou, ainda, que

$$E[Z_n X_l] = \sum_{j=1}^n h_j^{(n-1)} E[X_{n-j} X_l]$$

o que implica em

$$R_{Z,X}(n, l) = \sum_{j=1}^n h_j^{(n-1)} R_X(n-j, l), \quad (\text{A.5})$$

onde $R_{Z,X}$ representa a correlação de Z e X e R_X indica a autocorrelação de X .

Em um instante posterior, na predição do estado, tem-se que

$$Y_{n+1} = \sum_{j=1}^{n+1} h_j^{(n)} X_{n+1-j}. \quad (\text{A.6})$$

Considerando novamente a Condição de Ortogonalidade apresentada na Equação A.4, é possível obter que

$$R_{Z,X}(n+1, l) = \sum_{j=1}^{n+1} h_j^{(n)} R_X(n+1-j, l), \forall l = 0, 1, \dots, n-1. \quad (\text{A.7})$$

O número de equações envolvidas para determinar o número de estados cresce à medida que o número de observações aumenta. Uma forma de solucionar este crescimento é através da recursividade. Tomando a Equação A.7 e a definição de correlação e, em seguida, substituindo A.1, tem-se que

$$\begin{aligned} R_{Z,X}(n+1, l) &= E[Z_{n+1}X_l] = E[(a_n Z_n + W_n)X_l] \\ &= a_n E[Z_n X_l] + E[W_n X_l] \\ &= a_n R_{Z,X}(n, l), \end{aligned} \quad (\text{A.8})$$

assumindo que $E[W_n X_l] = 0$, o que equivale a dizer que o ruído do sistema é descorrelacionado com as observações passadas de X_n .

Tomando agora a Equação A.5 e substituindo A.2, tem-se que

$$\begin{aligned} R_{Z,X}(n, l) &= E[Z_n X_l] = E[(X_n - N_n)X_l] \\ &= E[X_n X_l] - E[N_n X_l] \\ &= E[X_n X_l] = R_X(n, l), \end{aligned} \quad (\text{A.9})$$

desde que o ruído de observação N_n seja descorrelacionado com as observações à priori de X , ou seja, $E[N_n X_l] = 0$. Em seguida, substituindo a Equação A.7 em A.8, obtém-se

$$\begin{aligned} a_n R_X(n, l) &= R_{Z,X}(n+1, l) = \sum_{j=1}^{n+1} h_j^{(n)} R_X(n+1-j, l) \\ &= h_1^{(n)} R_X(n, l) + \sum_{j=2}^{n+1} h_j^{(n)} R_X(n+1-j, l) \end{aligned}$$

Da Equação A.9, tem-se que $R_{Z,X}(n, l) = R_X(n, l)$, logo

$$(a_n - h_1^{(n)}) R_{Z,X}(n, l) = \sum_{j'=1}^n h_{j'+1}^{(n)} R_X(n-j', l).$$

Por fim, tem-se que

$$R_{Z,X}(n, l) = \sum_{j'=1}^n \frac{h_{j'+1}^{(n)}}{(a_n - h_1^{(n)})} R_X(n - j, l). \quad (\text{A.10})$$

O conjunto de equações apresentados em A.5 e A.10 são idênticos se

$$h_j^{(n-1)} = \frac{h_{j'+1}^{(n)}}{(a_n - h_1^{(n)})}, \text{ ou seja, } h_{j'+1}^{(n)} = (a_n - h_1^{(n)})h_j^{(n-1)}, \text{ para } j = 1, \dots, n. \quad (\text{A.11})$$

O objetivo agora é encontrar $h_1^{(n)}$.

A.2 Otimização da Predição

Esta seção tem como objetivo otimizar a predição de um processo observado, minimizando o erro médio quadrático de predição $E[\varepsilon_{n+1}^2]$. Inicialmente, substituindo a Equação A.11 em A.6, tem-se

$$\begin{aligned} Y_{n+1} &= h_1^{(n)} X_n + \sum_{j=2}^{n+1} h_j^{(n)} X_{n+1-j} = h_1^{(n)} X_n + \sum_{j'=1}^n h_{j'+1}^{(n)} X_{n-j'} \\ &= h_1^{(n)} X_n + \sum_{j'=1}^n (a_n - h_1^{(n)}) h_j^{(n-1)} X_{n-j'}. \end{aligned} \quad (\text{A.12})$$

Retomando a Equação A.3 e substituindo em A.12, tem-se, por fim, que

$$\begin{aligned} Y_{n+1} &= h_1^{(n)} X_n + (a_n - h_1^{(n)}) Y_n \\ &= a_n Y_n + (X_n - Y_n) h_1^{(n)}, \end{aligned} \quad (\text{A.13})$$

onde Y_n é a predição no tempo n , $a_n Y_n$ é a predição no instante $n + 1$ baseada em informações passadas, $(X_n - Y_n)$ é a inovação dada pela diferença entre os valores *a priori* da predição e da observação, denominada daqui em diante por I_n . O ganho de inovação usado para corrigir $a_n X_n$ é aqui representado por $h_1^{(n)}$, denominado agora por k_n . Reescrevendo Equação A.13, tem-se

$$Y_{n+1} = a_n Y_n + k_n I_n. \quad (\text{A.14})$$

Reescrevendo a equação do fator de inovação, tem-se

$$I_n = X_n - Y_n = (Z_n + N_n) - Y_n = (\varepsilon_n + Y_n) + N_n - Y_n \Rightarrow I_n = \varepsilon_n + N_n,$$

e tomando a equação do erro de predição no instante $n+1$ e substituindo as Equações A.1 e A.14, tem-se, finalmente, que

$$\begin{aligned}\varepsilon_{n+1} &= a_n Z_n + W_n - (a_n Y_n + k_n I_n) \\ &= a_n (Z_n - Y_n) + W_n - k_n (\varepsilon_n + N_n) \\ &= \varepsilon_n (a_n - k_n) + W_n - k_n N_n,\end{aligned}\tag{A.15}$$

assumindo a condição inicial $\varepsilon_0 = Z_0$. Constata-se que $E[\varepsilon_n] = 0, \forall n$, pois X_0, W_n e N_n têm média zero.

Aplicando o operador esperança na Equação A.15, tem-se a equação recursiva do erro médio quadrático (mse - *mean square error*) de predição, dada por

$$E[\varepsilon_{n+1}^2] = (a_n - k_n)^2 E[\varepsilon_n^2] + E[W_n^2] + k_n^2 E[N_n^2]\tag{A.16}$$

com condição inicial $E[\varepsilon_0^2] = E[Z_0^2]$. O objetivo agora é encontrar o valor de k_n que minimize $E[\varepsilon_{n+1}^2]$. Para isto, toma-se a derivada da Equação A.16 em relação a k_n , dada por

$$0 = -2(a_n - k_n)E[\varepsilon_n^2] + 2k_n E[N_n^2],$$

donde se obtém que

$$(a_n - k_n)E[\varepsilon_n^2] = k_n E[N_n^2].\tag{A.17}$$

Por fim, o ganho de inovação é recursivamente calculado por

$$k_n = \frac{a_n E[\varepsilon_n^2]}{E[\varepsilon_n^2] + E[N_n^2]},\tag{A.18}$$

Substituindo a Equação A.17 em A.16, tem-se a otimização da predição dada por

$$E[\varepsilon_{n+1}^2] = a_n(a_n - k_n)E[\varepsilon_n^2] + E[W_n^2].\tag{A.19}$$

Quando aninhadas, as Equações A.18, A.19 e A.14 representam o algoritmo de filtragem ótima de Kalman. É importante observar que o algoritmo requer o conhecimento a priori da estrutura do sinal a ser filtrado, ou seja, Y_o , a_n , $E[N_n^2]$ e $E[W_n^2]$.

O simulador apresentado no Anexo B tem implementado o algoritmo de filtragem ótima unidimensional de Kalman na função `KALMAN_FILTER`, cuja sintaxe é

$$[Y_{n+1}, E[\varepsilon_{n+1}^2]] = \text{Kalman_Filter}(X_n, Y_n, E[\varepsilon_n^2], E[W_n^2], E[N_n^2]).$$

ANEXO B – Simulador Desenvolvido

O simulador utilizado neste trabalho nos Capítulos 3 e 4, foi inteiramente desenvolvido em linguagem interpretada do Matlab®. O objetivo de desenvolver um simulador para um robô móvel tipo uniciclo é o de facilitar e agilizar os testes dos algoritmos de controle propostos nesta dissertação, assim como as estratégias de evasão de obstáculos. Adicionalmente, através do simulador é possível realizar simulações de múltiplos robôs utilizando sensores laser de varredura em todos eles, o que atualmente não é de possível execução no laboratório em que foi desenvolvido este projeto, visto que não há disponibilidade de tais equipamentos em quantidade (só há um sensor laser de varredura, embora hajam dois Pioneer 2DX).

O simulador está dividido em quatro partes, a saber: parte gráfica, sistema sensorial, estratégias de desvio de obstáculos e de formação e algoritmos de controle. A parte gráfica envolve basicamente a parte visual do simulador, a qual contempla o robô Pioneer 2DX e o sensor de varredura laser SICK LMS200, ambos em suas dimensões reais. Adicionalmente, foram implementadas rotinas para definir e desenhar o ambiente de navegação (espaço de trabalho de um robô móvel ou de uma formação), para inserir obstáculos de formas livres (definidas pelo usuário) e para indicar a posição do alvo a ser atingido (para o caso de um controle de posição). Quanto ao sistema sensorial, o simulador dispõe do sensor de varredura laser, o qual foi largamente utilizado neste trabalho. A simulação deste sensor leva em conta o ambiente de navegação e a disposição de possíveis obstáculos e de outros robôs móveis (desde que estejam devidamente identificados). As estratégias de desvio de obstáculo para um único robô ou para uma formação, apresentadas no texto desta dissertação, foram implementadas em simulação. Tais estratégias utilizam os dados do ambiente providos pelo sistema sensorial aqui simulado. Por fim, os algoritmos de controle descentralizado de formação e de posição e orientação foram implementados, a fim de executar, de forma semelhante, as tarefas dadas a um robô real.

Cada parte do simulador é explicada em detalhes nas seções subseqüentes, onde são mencionadas as funções desenvolvidas e seus respectivos cabeçalhos.

B.1 Parte Gráfica

ROBOT_DIMENSION: define as dimensões do robô Pioneer 2DX. As dimensões reais deste robô são os valor *default* aqui adotados. Entretanto, em algumas aplicações, para melhor visualização, torna-se necessário ampliar ou reduzir estas dimensões. Para isto, foi inserido um fator da escala.

Sintaxe: `Robot_Dimension(Scale)`

ROBOT_POSITION_2D: plota no ambiente um robô bi-dimensional definido por `ROBOT_DIMENSION`, para simulação com vista de cima (topo). Com base nos sinais de controle enviados, na pose atual e no modelo cinemático do robô móvel tipo unicycle, determina a pose a posteriori. A função retorna um objeto *robô*, o qual é utilizado para definir quando o mesmo deve aparecer (ou não) na cena de simulação. A função `DELETE` deve ser utilizada para apagar o robô da cena.

Sintaxe: `[x,y, ψ ,Robot] = Robot_Position_2D($\nu,\omega,x,y,\psi,color$)`

ROBOT_POSITION: de forma similar a `ROBOT_POSITION_2D`, plota no ambiente um robô tridimensional definido por `ROBOT_DIMENSION`, para simulações onde exista alteração da vista da cena.

Sintaxe: `[x,y, ψ ,Robot] = Robot_Position($\nu,\omega,x,y,\psi,color$)`

ROBOT_OBSTACLE: nesta função são definidas as dimensões do espaço de trabalho, ou seja, os limites máximos de navegação dos robôs, assim como é definido o número de obstáculos a ser inserido no ambiente. Tais obstáculos têm sua forma e posição definidas pelo usuário e, posteriormente, são exibidos no ambiente. A função retorna o número de obstáculos e as coordenadas, no plano **XY**, dos mesmos.

Sintaxe: `[Obstacle,N] = Robot_Obstacle`

ROBOT_REGION_GEN: a função `ROBOT_OBSTACLE` retorna as coordenadas dos vértices dos obstáculos desenhados. Entretanto, para interação do robô móvel com o meio, é necessário subdividir as retas que unem tais vértices em intervalos menores. A estes intervalos dá-se o nome de sensibilidade (**Sensitivity**). Uma vez calculadas as coordenadas **XY** de cada par de vértices, estas são colocadas em um vetor e retornadas pela função. A função deve ser utilizada individualmente para cada obstáculo, o qual terá suas coordenadas inseridas no final do vetor que define o ambiente (**Environment**).

Sintaxe: `Environment = Robot_Region_Gen(Obstacle,Environment,Sensitivity)`

ROBOT_TARGET: com o objetivo de ilustrar o alvo a ser atingido pelo robô móvel no ambiente, esta função recebe as coordenadas da posição a ser atingida, plota uma bandeira nesta posição e retorna o objeto *alvo*. A função **DELETE** deve ser utilizada para apagar da cena a bandeira indicativa do alvo.

Sintaxe: `Target = ROBOT_TARGET(x,y,Scale)`

B.2 Sistema Sensorial

ROBOT_LASER_VALUES: obtém os valores de distância do sensor de varredura laser em função da pose do robô e do ambiente ao redor de si. Nesta função é calculada a distância Euclidiana entre as coordenadas do robô móvel e de cada ponto no espaço de trabalho, cujas coordenadas estão armazenadas no vetor **Environment** resultante da função **ROBOT_REGION_GEN**. De forma similar, é computado o ângulo entre a pose do robô e os objetos no ambiente. Em seguida, é feito um mapa polar do ambiente, e, por fim, faz-se a busca pela distância mínima para cada ângulo de 0° a 180° (em intervalos de 1°), considerando a orientação do robô móvel no ambiente. A função retorna um vetor de 181 posições correspondente às medidas de distância para o intervalo angular $[0^\circ, 180^\circ]$. O alcance máximo das medidas do sensor laser simulado (**LaserMaxRange**) é definido nesta função. Às medidas de distância que superam o alcance estabelecido, naquela posição angular, é atribuído **LaserMaxRange**. No simulador do sensor laser, foi inserido um ruído branco ($\mu = 0m$ e $\sigma = 25 \times 10^{-6}m^2$) nos valores das medições, em correspondência com a realidade.

Sintaxe: `M = Robot_Laser_Values(x,y, ψ ,Environment)`

ROBOT_LASER: de posse de **M**, a função exibe a varredura realizada pelo sensor, a fim de representar o alcance “visual” do robô móvel e sua interação com o ambiente. Adicionalmente, é desenhado um prisma sobre o robô, para ilustrar o sensor laser a bordo. A função recebe como parâmetro a pose do robô em relação ao referencial inercial, e retorna um objeto “Laser”. A função **DELETE** deve ser utilizada para apagar da cena o sensor laser a bordo do robô.

Sintaxe: `Laser = Robot_Laser(M,x,y, ψ ,Scale)`

B.3 Estratégia de Desvio de Obstáculos

ROBOT_FIND_ROTATE_ANGLE_GAMMA: retorna o ângulo de giro γ do alvo real para criar o alvo virtual, conforme estabelecido pelo algoritmo de desvio tangencial modificado descrito na Seção 2.2. A função recebe como parâmetros o vetor de medidas do sensor laser de varredura, o erro de orientação do robô no ambiente e a distância que define a região de segurança (d_{obs}), e fornece a distância mínima a um obstáculo (d_{min}), e, por fim, o ângulo de giro γ . SN_{obs} é um *flag* booleano que indica a presença ou não de obstáculos no ambiente.

Sintaxe: $[\gamma, \beta, d_{min}, SN_{obs}] = \text{Robot_Find_Rotate_Angle_Gamma}(\mathbf{M}, d_{obs}, \alpha)$

ROBOT_TANGENTIAL_ESCAPE: retorna as coordenadas do alvo virtual com base no ângulo de giro γ resultante da função **ROBOT_FIND_ROTATE_ANGLE_GAMMA**. Caso $SN_{obs} = 1$, há obstáculo no ambiente de navegação para $d_{min} < d_{obs}$. Neste caso, a função recebe a pose atual do robô móvel, as coordenadas do alvo, o ângulo de giro γ e a distância mínima entre o robô e um obstáculo, e, finalmente, determina as coordenadas do alvo virtual. O ângulo β indica se o obstáculo está à direita ($\beta < 0$) ou à esquerda ($\beta \geq 0$) do robô. Caso $SN_{obs} = 0$, a função retorna o próprio alvo real, pois não há obstáculos nas proximidades do robô móvel. O vetor de medidas \mathbf{M} da varredura laser também é um parâmetro de entrada, pois o mesmo é utilizado na implementação do seguidor de paredes e no cálculo do ângulo de ajuste γ_{adj} , o qual corrige o ângulo de giro γ .

Sintaxe: $[X_v, Y_v] = \text{Robot_Tangential_Escape}(x, y, \psi, X_d, Y_d, \gamma, \beta, d_{min}, SN_{obs}, \mathbf{M})$

B.4 Estratégias Aplicadas ao Controle de Formação

ROBOT_FIND_PATTERN: implementa a estratégia de busca do padrão montado sobre o robô líder, a qual foi proposta na Seção 3.3, utilizando o perfil de medidas do sensor de varredura laser. A função retorna as variáveis de formação $[\rho_{LF}, \beta_{LF}, \theta_{LF}]^T$, que indicam a posição e orientação do robô líder relativas ao robô seguidor, bem como os ângulos em que foram detectados o início e o fim do padrão no perfil laser, definidos, respectivamente, por β_1 e β_2 .

Sintaxe: $[\rho_{LF}, \beta_{LF}, \theta_{LF}, \beta_1, \beta_2] = \text{Robot_Find_Pattern}(\mathbf{M}, \rho_{LF}, \beta_{LF}, \theta_{LF}, \beta_1, \beta_2)$

ROBOT_CHANGE_BETAD: altera o valor do ângulo de formação desejado, conforme discutido na Seção 4.2. As medidas de distância provenientes do sensor laser

são aqui utilizadas para verificar a presença de obstáculos no ambiente. Os possíveis obstáculos devem se encontrar no intervalo de medidas $[0^\circ, \beta_1)$, à direita do robô seguidor, e/ou de $(\beta_2, 180^\circ]$, à sua esquerda, visto que no intervalo $[\beta_1, \beta_2]$ encontra-se o robô líder identificado por seu padrão. Caso exista algum obstáculo a uma distância inferior à distância de segurança, é gerada uma força fictícia (cuja intensidade máxima pode ser alterada), e, por conseguinte, uma alteração no valor de β_{LFd} . Caso contrário, a função retorna o valor inicialmente estabelecido para β_{LFd} .

Sintaxe: $\beta_{LFd} = \text{Robot_Change_Betad}(\mathbf{M}, \beta_1, \beta_2, d_{obs}, \rho_{LFd}, \beta_{LFd})$

B.5 Controladores

ROBOT_POSITION_CONTROL: fornece os sinais de controle necessários para atingir um alvo no plano \mathbf{XY} , e ali permanecer. O controlador aqui implementado é aquele descrito na Seção 1.3.1. A função recebe como parâmetros a pose corrente do robô móvel e a posição desejada (alvo virtual dado por **ROBOT_TANGENTIAL_ESCAPE**). Os ganhos do controlador (ν_{max} e k_ω) podem ser alterados nesta função.

Sintaxe: $[\nu, \omega] = \text{Robot_Position_Control}(x, y, \psi, X_d, Y_d)$

ROBOT_ORIENTATION_CONTROL: re-orienta o robô móvel, para que este possa atingir o alvo com uma orientação desejada. A função retorna os sinais de controle necessários para efetuar o giro do robô sobre seu próprio eixo. Como parâmetros de entrada a função recebe a pose atual do robô, as coordenadas do alvo desejado e a orientação desejada ψ_d .

Sintaxe: $[\nu, \omega] = \text{Robot_Orientation_Control}(x, y, \psi, X_d, Y_d, \psi_d)$

ROBOT_FOLLOWING_CONTROL: implementado no robô seguidor, esta função fornece os sinais de controle para que este robô seja capaz de estabelecer a formação linear de estrutura definida por ρ_{LFd} e β_{LFd} , conforme discutido na Seção 3.2. O ângulo de formação desejado β_{LFd} pode, porventura, ter sido alterado por **ROBOT_CHANGE_BETAD** devido à presença de obstáculos dentro da região de segurança da formação. A função recebe como parâmetro a pose atual do robô líder através das variáveis de formação (calculadas em **ROBOT_FIND_PATTERN**) e a velocidade linear estimada do robô líder, ν_L .

Sintaxe: $[\nu_F, \omega_F] = \text{Robot_Following_Control}(\rho_{LF}, \beta_{LF}, \theta_{LF}, \nu_L, \rho_{LFd}, \beta_{LFd})$

B.6 Exemplo

O script a seguir ilustra o controle descentralizado com desvio de obstáculos para uma formação líder-seguidor semi-rígida de robôs móveis.

```
clear all
close all
clc

%%%% Senbilidade do sistema - distância entre os pontos %%%%
Sense = 0.01;
T = 0.1; %% Período de amostragem

%%%% Valores desejados da formação
rhod_ms = 1.0;
betad_ms = pi/2;
betad_ms_temp = betad_ms;

%%% Condições iniciais do robô líder
xmRobot = 0;    % [m]
ymRobot = 0;    % [m]
pmRobot = 0;    % [graus]
um = 0;
wm = 0;

%%%% Condições iniciais do sistema seguidor
% Condições iniciais do robô seguidor

xsRobot = -sin(betad_ms)*rhod_ms;    % [m]
ysRobot = cos(betad_ms)*rhod_ms;    % [m]
psRobot = 0;                        % [graus]
us = 0;
ws = 0;

%%% Posição desejada a ser alcançada pelo robô líder
Xdm = 10;
Ydm = 0;

%%% Valores iniciais da formação
rho_ms = sqrt((xmRobot-xsRobot)^2+(ymRobot-ysRobot)^2);
beta_ms = atan2(ymRobot-ysRobot,xmRobot-xsRobot) + pi/2;
theta_ms = 0;
rho_ms_ant = rho_ms;

% Sistema de Seguimento
rho_ms_aux = rho_ms;
beta_ms_aux = beta_ms;
theta_ms_aux = theta_ms;
beta1 = 80;
beta2 = 100;

%% Padrão montado sobre o robô líder
Leader_Pattern = [0.1*cos(-pi/2:pi/180:pi/2);0.1*sin(-pi/2:pi/180:pi/2)];

%% Inicialização do Filtro de Kalman
%% Parâmetros Kalman Unidimensional
%% Parâmetros para Estimação da Velocidade Linear do Líder
MSEum = 0.0; VARWum = 0.000001; VARNum = 0.00025; Yum = 0.1;

%% Parâmetros para Filtragem das Variáveis de Formação
MSEr = 0.0; VARWr = 0.000001; VARNr = 0.0000002; Yr = rhod_ms; %% rho
MSEb = 0.0; VARWb = 0.000001; VARNb = 0.25*pi^2/180^2; Yb = betad_ms; %% beta
MSEt = 0.0; VARWt = 0.000001; VARNt = 16.0*pi^2/180^2; Yt = 0.0; %% theta

%% Estimação de velocidades
um_est = 0.0; umsf = 0.0;

%% Condições iniciais do sensor laser e distância da região de segurança
```

```

LaserMaxRange = 5;
dist_obst = 0.5;

%% Programa Principal
figure(1)
Robot_Dimension
[xmRobot,ymRobot,pmRobot,RobotM] = robot_position(um,wm,xmRobot,ymRobot,pmRobot,'b');
[xsRobot,ysRobot,psRobot,RobotS] = robot_position(us,ws,xsRobot,ysRobot,psRobot,'r');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Definição do Ambiente %%%%%%%%%
[Obstacle,N] = Robot_Obstacle;

for nob = 1:N
    if nob == 1
        Entorno = Robot_Region_Gen(Obstacle{nob},Sense);
    else
        Entorno = Robot_Region_Gen(Obstacle{nob},Sense,Entorno);
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Fim da definição do ambiente %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Target = Robot_Target(Xdm,Ydm,0.001);

%%% Vista do sistema
view(0,90)
cameratoolbar

rho = sqrt((Xdm - xmRobot)^2 + (Ydm - ymRobot)^2);

while rho > 0.05

    Robot_Laser_dim = [xmRobot 0; 0 ymRobot]*ones(size(Leader_Pattern)) + ...
        [cos(pmRobot) -sin(pmRobot); sin(pmRobot) cos(pmRobot)]*Leader_Pattern;
    Robot_Leader = Robot_Region_Gen(Robot_Laser_dim,Sense);
    Entorno_Global = [Entorno Robot_Leader];

    delete(RobotS,RobotM)

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%% ROBÔ LÍDER %%%%%%%%%
    rho = sqrt((Xdm - xmRobot)^2 + (Ydm - ymRobot)^2); % Erro de posição do robô [m]
    theta = atan2(Ydm - ymRobot,Xdm - xmRobot); % Ângulo entre robô e destino [graus]
    alpha = theta - pmRobot; % Erro de orientação do robô em relação ao ponto destino [graus]

    Robot_Laser_Datos_M = Robot_Laser_Values(xmRobot,ymRobot,pmRobot,Entorno);
    [gamma,beta,dist_min,SN_obst] = Robot_Find_Rotate_Angle_Gamma(Robot_Laser_Datos_M,dist_obst,alpha);
    [Xtemp,Ytemp] = Robot_Tangential_Escape(xmRobot,ymRobot,pmRobot,Xdm,Ydm,...
        gamma,beta,dist_min,SN_obst,Robot_Laser_Datos_M);

    [um,wm] = Robot_Position_Control(xmRobot,ymRobot,pmRobot,Xtemp,Ytemp,gamma);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%% ROBÔ SEGUIDOR %%%%%%%%%
    %%
    %% Buscando o padrão por meio das medidas laser
    Robot_Laser_Datos = Robot_Laser_Values(xsRobot,ysRobot,psRobot,Entorno_Global);
    [rho_ms,beta_ms,theta_ms,beta1,beta2] = Robot_Find_Pattern(Robot_Laser_Datos,...
        rho_ms,beta_ms,theta_ms,beta1,beta2);

    if (abs(rho_ms - rho_ms_aux) > 1.0 | abs(beta_ms - beta_ms_aux) > pi/4.0)
        rho_ms = rho_ms_aux;
        beta_ms = beta_ms_aux;
        theta_ms = theta_ms_aux;
    else
        rho_ms_aux = rho_ms;
        beta_ms_aux = beta_ms;
        theta_ms_aux = theta_ms;
    end

    betad_ms = Robot_Change_Betad(Robot_Laser_Datos,beta1,beta2,dist_obst,rho_ms,betad_ms);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Filtro de Kalman %%%% Início %%%%
[Yum, MSEum] = kalman_filter(umsf, Yum, MSEum, VARWum, VARNum);
um_est = Yum;
[rho_ms,MSEr] = kalman_filter(rho_ms, Yr, MSEr, VARWr, VARNr);
Yr = rho_ms;
[beta_ms,MSEb] = kalman_filter(beta_ms, Yb, MSEb, VARWb, VARNb);
Yb = beta_ms;
[theta_ms,MSEt] = kalman_filter(theta_ms, Yt, MSEt, VARWt, VARNt);
Yt = theta_ms;
%%% Filtro de Kalman %%%% Fim %%%%

[us,ws] = Robot_Following_Control(rho_ms,beta_ms,theta_ms,um_est,rhod_ms,betad_ms);

%%% Estimação da Velocidade Linear do Robô Líder %%%% Início %%%%
ang_min_ms = beta_ms - theta_ms;

if (abs(ang_min_ms) < pi/180.0)
    if (ang_min_ms > 0.0)
        ang_min_ms = 2*pi/180.0;
    else
        ang_min_ms = -2*pi/180.0;
    end
end

um_est = ((rho_ms - rho_ms_ant)/T + us*sin(beta_ms))/sin(ang_min_ms);
umsf = um_est; %% sem filtrar

rho_ms_ant = rho_ms;
%%% Estimação da Velocidade Linear do Robô Líder %%%% Fim %%%%

figure(1)
[Rlaser_M] = robot_laser(Robot_Laser_Datos_M*0,xmRobot,ymRobot,pmRobot);
[Rlaser_S] = robot_laser(Robot_Laser_Datos,xsRobot,ysRobot,psRobot);
[xmRobot,ymRobot,pmRobot,RobotM] = robot_position(um,wm,xmRobot,ymRobot,pmRobot,'b');
[xsRobot,ysRobot,psRobot,RobotS] = robot_position(us,ws,xsRobot,ysRobot,psRobot,'r');

%% Circunferência de referência da formação rígida
SafeRegion_M = patch(xmRobot + dist_obst*cos(0:pi/45:2*pi),...
    ymRobot + dist_obst*sin(0:pi/45:2*pi),...
    ones(1,91)*0.05,[1 1 0],'FaceAlpha',0.2);
SafeRegion_S = patch(xsRobot + (dist_obst+rhod_ms)*cos(0:pi/45:2*pi),...
    ysRobot + (dist_obst+rhod_ms)*sin(0:pi/45:2*pi),...
    ones(1,91)*0.05,[0 1 1],'FaceAlpha',0.2);

pause(0.001)
delete(Rlaser_S,Rlaser_M,MS,SafeRegion_M,SafeRegion_S)
end

delete(Target)

```

ANEXO C – Plataforma Experimental - Pioneer 2DX

Pioneer é uma família de robôs móveis a rodas de pequenas dimensões e dotada de um certo grau de inteligência, cuja arquitetura foi desenvolvida por Kurt Konolige, da Empresa SRI International e da Universidade de Stanford. Pioneer é um robô do tipo “*plug and play*”, o qual possui todos os componentes básicos de sensoramento e navegação em ambientes do mundo real. Especificamente, ele é provido de rodas de tração diferencial, codificadores óticos (*encoders*) de posição e velocidade, baterias internas e sensores integrados e acessórios, os quais são gerenciados por um microcontrolador a bordo.

O Pioneer apresenta também uma variedade de dispositivos de expansão e portas de entrada/saída (I/O) para acessórios opcionais, incluindo um barramento I/O endereçável para conexão de 16 dispositivos, duas portas padrão RS-232, oito portas digitais de I/O e cinco conversores A/D. Todos estes itens são acessados através da interface *Pioneer 2 Operating System* - P2OS, ou Sistema Operacional do Pioneer 2. Com o computador opcional a bordo, o Pioneer 2 pode ter ainda quatro portas RS-232 e um barramento PCI.

Projetado para navegar em superfícies planas, o Pioneer é provido de um par de rodas de borracha com tração diferencial e uma roda castor traseira utilizada para balancear sua plataforma. Além disso, tal família de robôs dispõe de um microcontrolador Siemens 88C166 200MHz de alto desempenho e placas independentes para controle dos motores e dos sonares, que o torna versátil em uma infinidade de aplicações.

Para interações com o ambiente, o Pioneer é equipado com codificadores óticos de alta precisão (9850 pulsos por revolução), que lhe fornece uma odometria eficiente, no que se refere ao posicionamento global durante a navegação, bem como fornece os valores correntes de suas velocidades linear e angular. Para auxiliar na detecção de objetos, o

Pioneer suporta dezesseis sonares, dispostos oito na parte dianteira e oito na parte traseira de sua plataforma.

No que se refere ao modo de operação, o Pioneer pode ser operado de quatro formas distintas, descritas a seguir:

Server Mode: A maioria dos usuários do Pioneer utiliza tal modo de operação, pois permite um acesso rápido e facilitado às funcionalidades do robô, enquanto se trabalha em softwares de mais alto nível no PC.

Standalone: Para a operação neste modo, o fabricante fornece os recursos para download, embora não forneça os programas do microcontrolador. Para experimentos a nível de microcontrolador é possível reprogramar as memórias flash RAM e ROM para sua operação direta.

Controle por Joystick e Self-Test: O Pioneer é provido de um software onboard que permite o controle e manipulação do robô mediante um joystick, bem como alguns programas de teste para exercitar o hardware e o software através de seu microcontrolador.

Apesar de suas dimensões físicas reduzidas, o Pioneer possui uma capacidade de inteligência impressionante, que o torna compatível com as máquinas de maior dimensão e muitas vezes mais caras.

Tendo sido projetado para navegar em ambientes semi-estruturados, tais como laboratórios, salas de aulas, galpões de fábrica e escritórios, o Pioneer é capaz de mapear um ambiente, encontrar um caminho para um dado objetivo e desempenhar outras tarefas de alto grau de complexidade, desde que devidamente programado.

Pesando em torno de 9kg, ele pode ser considerado leve. Contudo, o alumínio que constitui sua carcaça, de construção sólida e resistente, lhe permite suportar cargas mecânicas extraordinárias sobre sua estrutura, correspondentes a até 23kg de massa adicional.

O Pioneer é constituído basicamente de duas partes: a superfície de montagem e o corpo (ou carcaça). A superfície de montagem é o local onde são acoplados acessórios tais como sensor laser e sistemas de visão computacional. Estes normalmente são instalados nas proximidades da face dianteira, pois possíveis desbalanceamentos do robô podem ser compensados pelo conjunto de baterias inserido na parte traseira do corpo (ou carcaça) do Pioneer. Esta parte do robô é o local onde estão alojados os *drivers* do motor, os

componentes eletrônicos, o *array* de sonares e o conjunto de baterias. Há, ainda, espaço suficiente para um computador de bordo, *modem* de rádio e Ethernet, e placa de sistema de visão, dentre outros.

O software utilizado em conexão com os robôs Pioneer devem utilizar sua biblioteca ARIA. ARIA é a abreviação de *ActivMedia Robotics Interface for Application*, o qual é um software orientado a objeto para programação de aplicações de controle de robôs inteligentes. Escrito na linguagem C++, ARIA é um software cliente desenvolvido para facilitar/otimizar o acesso e o gerenciamento de tarefas executadas no robô servidor, como também por acessórios nele instalados, tais como sensores (câmaras e laser) e efetadores (motores e garras). A versatilidade e flexibilidade faz do ARIA uma excelente base de programação de aplicações robóticas em alto nível.

ARIA pode ser programado para rodar em *multi-threaded* ou *single-threaded*, ou seja, programação paralela ou sequencial. Ela utiliza as características próprias de programação paralela do *Linux p Threads* e *WIN32 Threads*. Baseado nesta informação de lógica de programação, programas em ARIA podem ser escritos para simples controle de comandos do robô servidor em operações de navegação, bem como desenvolver ações inteligentes de alto nível. ARIA foi desenvolvido para aqueles que já possuem softwares de controle do robô e querem de forma rápida e fácil avaliá-la em uma ou mais plataformas de robôs móveis da ActivMedia.

O programa desenvolvido nesta Dissertação para ser executado no robô móvel Pioneer 2DX foi implementado em C++ utilizando a biblioteca ARIA, usando-se códigos distintos para os robôs líder e seguidor. A rotina implementada no robô seguidor é apresentada e comentada a seguir.

```

////////////////////////////////////
// Controlador de formação líder-seguidor baseada em sensoriamento laser
// Implementação no robô seguidor
//
// Alexandre S. Brandao
// INAUT - UFES - 16/12/2007
////////////////////////////////////

#include "Aria.h"
#include <ArRobot.h>
#include <iostream>
#include <ArTcpConnection.h>
#include "math.h"

#define PI 3.1415926

////-----*****READ_LASER*****-----////
// Realiza leitura do sensor laser e armazena dados em arquivo de texto
void read_laser(int *medidas, ArSick &sick, FILE *arq_laser){...}
}

////-----*****FIND_PATTERN*****-----////

```

```

// Busca a posição do robô líder com a orientação e ângulo de formação
void find_pattern(int *medidas, double *dx, double &rho_act, double &beta_act,
                 double &theta_act, int &beta1_act, int &beta2_act){...}

////-----*****CHANGE_BETAD*****-----////
// Verifica a situação do ambiente e na existência de algum obstáculo,
// a função retorna um valor modificado de beta_d, para o controle de formação
double change_betad(int *medidas, double rhod, double betad, int beta1, int beta2){...}

////-----*****INITIAL_COND*****-----////
// Encontra as condições iniciais do sistema
void initial_cond(ArSick &sick, int *medidas, double &rho, double &beta,
                 double &theta, FILE *arq_laser_ini){...}

////-----*****KALMAN_FILTER*****-----////
// Filtro de Kalman Unidimensional
double kalman_filter(double Xobs, double &Ypred, double &MSE, double VARW, double VARN){...}

////////////////////////////////////
// Programa Principal
double us = 0.0, ws, ur, wr;
int main(int argc, char **argv) {

    ArRobot robot;           // Adicionando objeto robô
    ArSonarDevice sonar;     // Inserindo objeto que torna possível a utilização dos sensores de ultra-som
    ArSick sick;             // Declarando objeto sensor de varredura laser

    ArKeyHandler keyHandler;

    robot.setTransVelMax(20);
    robot.setTransAccel(0);
    robot.setTransDecel(0);

    Aria::init();

    ArSimpleConnector connector(&argc, argv);
    connector.parseArgs();

    if (!connector.parseArgs() || argc > 1) {
        connector.logOptions();
        exit(1);
    }

    Aria::setKeyHandler(&keyHandler);
    robot.attachKeyHandler(&keyHandler);
    robot.addRangeDevice(&sonar);
    robot.addRangeDevice(&sick);

    if (!connector.connectRobot(&robot)){
        printf("Não foi possível conectar ao Robô... Saindo\n");
        Aria::shutdown();
        return 1;
    }

    connector.setupLaser(&sick);

    sick.runAsync();

    if (!sick.blockingConnect())
    {
        printf("Não foi possível conectar ao sensor laser SICK... Saindo\n");
        Aria::shutdown();
        return 1;
    }

    robot.comInt(ArCommands::ENABLE, 1);
    robot.comInt(ArCommands::SOUNDTOG, 0);

    robot.runAsync(true);
    sick.setMaxRange(5000);    // Alcance máximo do sensor laser dado em [mm]
}

```

```

ArPose posicao; // Posição atual do robô;

////////////////////////////////////
// Variáveis de formação
double rho, beta, theta, phi = 0.0; //, ang_min;

// Variáveis de início e fim do padrão
int beta1, beta2;

// Velocidade linear do líder a ser estimada [m/s]
double um = 0.0, umsf = 0.0;

// Velocidades linear e angular do robô seguidor dada em [m/s] e [graus/s], respectivamente
double us = 0.0, ws, ur, wr;

// Valores desejados
double rhod = 0.5, betad = PI/2.0, betad_temp;
betad_temp = betad;

// Variáveis auxiliares
double rho_aux = rhod, beta_aux = 0.0, theta_aux = 0.0, rho_ant = 0.0;

// Variáveis de error
double rhotil, betatil;

// Constantes do Controlador Desenvolvido
double KL1 = 0.35, KL2 = 0.5;

double KH1 = 0.25, KH2 = 0.25;

////////////////////////////////////
double T = 0.1;
int cont = 0;

int medidas[181]; // Medidas del Sensor Laser
double dx[180]; // Derivadas de las medições

// Inicialização Filtro Kalman
// Parâmetros Kalman Unidimensional
// Parâmetros para Estimação da Velocidade Linear do Mestre
double MSEum = 0.0, VARWum = 0.000001, VARNum = 0.00025, Yum = 0.1;

// Parâmetros para Filtragem das Variáveis de Formação
double MSEr = 0.0, VARWr = 0.000001, VARNr = 0.0000002, Yr = rhod; // rho
double MSEb = 0.0, VARWb = 0.000001, VARNb = 0.25*PI*PI/180.0/180.0, Yb = betad; // beta
double MSEt = 0.0, VARWt = 0.000001, VARNt = 16.0*PI*PI/180.0/180.0, Yt = 0.0; // theta

posicao = robot.getPose(); // Posição atual do robô

////////////////////////////////////
/// Determinando condições iniciais ////////// Início ///
initial_cond(sick, medidas, rho_aux, beta_aux, theta_aux, arq_laser_ini);

rho_ant = rho_aux;

Yr = rho_aux;
Yb = beta_aux;
Yt = theta_aux;
/// Determinando condições iniciais ////////// Fim ///

////////// Laço Principal de Controle //////////
while(cont < 1300)
{
    ////////// Busca do Padrão ////////// Início //////////
    // Varredura laser
    read_laser(medidas,sick,arq_laser);

    // Obtendo a derivada das medidas laser
    for (int i = 0;i < 180;i++)
        dx[i] = (double) (medidas[i+1] - medidas[i]);

    rho = sick.getMaxRange();

```

```

find_pattern(medidas,dx,rho,beta,theta,beta1,beta2);
rho /= 1000.0;
///////// Procura do Padrão /////////// Fim   ////

/////////
// Modificação do ângulo de formação para evasão de obstáculos em uma formação semi-rígida
betad_temp = change_betad(medidas,rhod,betad,beta1,beta2);

///////// Filtro de Kalman /////////// Início ////
um = kalman_filter(um, Yum, MSEum, VARWum, VARNum);

rho = kalman_filter(rho, Yr, MSEr, VARWr, VARNr);
beta = kalman_filter(beta, Yb, MSEb, VARWb, VARNb);
theta = kalman_filter(theta, Yt, MSET, VARWt, VARNt);
///////// Filtro de Kalman /////////// Fim   ////

rhotil = rhod - rho;
betatil = betad_temp - beta;

/////////
// Projeto de Controlador por Lyapunov e Cinemática Inversa
us = (um*sin(beta-theta) - KH1*tanh(rhotil))/sin(beta);
ws = (KH1*tanh(rhotil)*cos(beta) + um*sin(theta))/rho/sin(beta) - KH2*tanh(betatil);

// Enviadas ao Pioneer
robot.setVel(us*1000); // [mm/s]
robot.setRotVel(ws*180.0/PI); // [grados/s]

// Recebidas do Pioneer
ur = robot.getVel();
wr = robot.getRotVel();

/////////
///////// Estimção da velocidade linear do líder /////////// Início ////
um = ((rho - rho_ant)/T + us*sin(beta))/sin(beta-theta);
umsf = um; // sin filtrar

rho_ant = rho;
///////// Estimção da velocidade linear do líder /////////// Fim   ////

posicao = robot.getPose();
cont++;

ArUtil::sleep(100);
}
robot.disconnect();
Aria::shutdown();
return 0;
}

```

Por outro lado, o programa instalado no robô líder, que implementa a estratégia de desvio tangencial utilizando os sensores ultra-sônicos, é descrito e comentado a seguir.

```

/////////
// Controlador de formação líder-seguidor baseada em sensoramento laser
// Implementação no robô líder: Controlador de Posição sem Orientação
// com Desvio Tangencial Modificado utilizando sonsares
//
// Alexandre S. Brandão
// INAUT - UFES - 16/12/2007
/////////

#include "Aria.h"
#include <ArRobot.h>
#include <iostream>

using namespace::std;
#include "math.h"
#define PI 3.14159265

```

```

ArRobot robot;
ArSonarDevice sonar;

////////////////////////////////////
////-----*****FIND_BETA*****-----////
// Busca da medida mais próxima retornando o ângulo gamma
// a ser utilizado na rotação do alvo para sensor frontal ultra-som
double find_beta(int *medidas, int &dist_min, int &dist_90g){
    int i_min = 4;
    double beta = 0.0, angulo[8] = {90.0,50.0,30.0,10.0,-10.0,-30.0,-50.0,-90.0};

    for(int i = 0; i < 8; i++){
        if (medidas[i] <= dist_min){
            dist_min = medidas[i];
            beta = angulo[i];
            i_min = i;
        }

        if (beta >= 0)
            dist_90g = medidas[i_min-4];
        else
            dist_90g = medidas[i_min+4];

    }

    return beta;
}

////-----*****FIND_GAMMA*****-----////
// Busca o ângulo de desvio para o caso em que exista um objeto
// a uma distancia menor que um valor especificado
bool find_gamma(int dist_min, int dist_90g, const int dist_obst, double beta, double alpha, double &gamma){
    if (dist_min < dist_obst){
        if (beta < 0.0)
            gamma = 90.0 + beta - alpha;
        else
            gamma = -90.0 + beta - alpha;

        return true; // Condição para indicar a existência de obstáculo
    }

    return false; // Condição para indicar a não existência de obstáculo
}

////-----*****TAN_ESCAPE*****-----////
// Desvio Tangencial ajustado por seguimento de paredes
void tan_escape(ArRobot &robot, ArPose Xc, ArPose Xd, ArPose &Xtmp,
                double beta, double &gamma, int dist_min, int *medidas){
    // Existe um obstáculo no caminho do robô e deve ser evitado tangencialmente
    double rho, psi, theta, alpha;

    // Xc = Posicao atual do robo
    // Xd = Posicai desejada
    rho = Xc.findDistanceTo(Xd)/1000.0; // Erro de posicao do robo [m]

    psi = robot.getTh(); // [graus] orientação do robô
    theta = Xc.findAngleTo(Xd); // [graus]
    alpha = theta - psi; // [graus] erro de orientação do robô com relação ao ponto destino

    // Variáveis para criação do destino virtual
    double xvirt, xtemp, yvirt, ytemp;

    // Alvo virtual rotacionado
    xvirt = dist_min * cos((double)theta * PI / 180.0);
    yvirt = dist_min * sin((double)theta * PI / 180.0);

    // Obtendo alvo virtual
    xtemp = xvirt * cos(gamma * PI / 180.0) - yvirt * sin(gamma * PI / 180.0) + Xc.getX();
    ytemp = xvirt * sin(gamma * PI / 180.0) + yvirt * cos(gamma * PI / 180.0) + Xc.getY();

    // Setando distância do alvo virtual em relação à origem absoluta
    Xtmp.setX(xtemp);
    Xtmp.setY(ytemp);
}

```

```

////-----*****POS_CTRL*****-----////
// Controlador de Posição Final
void pos_ctrl(ArRobot &robot, ArPose Xc, ArPose Xd, ArPose Xtmp, double gamma, FILE *arq_action_ctrl){
    double kv = 0.15, kw = 0.20;

    // Velocidades linear e angular do robo [m/s] e [graus/s]
    double v, w, vr, wr;
    double rho, psi, theta, alpha;

    // Xc = Posicao atual do robo
    // Xd = Posicao desejada
    rho = Xc.findDistanceTo(Xtmp)/1000.0; // Erro de posição do robô [m]

    psi = robot.getTh(); // [graus] orientação do robô

    theta = Xc.findAngleTo(Xd); // [graus]
    alpha = theta - psi + gamma; // [graus] erro de orientação do robô com relação ao ponto destino

    if (fabs(alpha) > 180.0)
        if (alpha >= 0.0)
            alpha -= 360.0;
        else
            alpha += 360.0;

    v = (kv * tanh(rho) * cos(alpha * PI / 180.0)) * 1000.0 ; //[mm/s]
    w = (kw * (alpha * PI / 180.0) + kv * (tanh(rho)/(rho)) * sin(alpha * PI / 180.0) *
        cos(alpha * PI / 180.0)) * 180.0 / PI; // [graus/s]

    // Enviadas ao Pioneer
    robot.setVel(v);
    robot.setRotVel(w);

    // Recebidas do Pioneer
    vr = robot.getVel();
    wr = robot.getRotVel();
}

////-----*****MAIN*****-----////
int main(int argc, char **argv)
{
    robot.setTransVelMax(20);
    robot.setTransAccel(0);
    robot.setTransDecel(0);

    ArKeyHandler keyHandler;

    Aria::init();

    ArSimpleConnector connector(&argc, argv);

    if (!connector.parseArgs() || argc > 1){
        connector.logOptions();
        exit(1);
    }
    Aria::setKeyHandler(&keyHandler);
    robot.attachKeyHandler(&keyHandler);
    robot.addRangeDevice(&sonar);

    if (!connector.connectRobot(&robot)){
        printf("Could not connect to robot... exiting\n");
        Aria::shutdown();
        return 1;
    }
    robot.comInt(ArCommands::ENABLE, 0);
    robot.comInt(ArCommands::SOUNDTOG, 0);
    robot.runAsync(true);

    // Referentes às posições do robô
    ArPose desttmp; // Posição temporária do ponto de destino

```

```

    ArPose posicao; // Posição atual do robô;
    ArPose destino; // Posição final desejada [mm, mm , graus]

    // Variáveis do controlador de posição
    double T = 0.1; // Período de amostragem
    double rho, alpha;

    // Variáveis do algoritmo de desvio de obstáculos
    double theta, phi, gamma, beta;

    // Medidas Referentes ao Controlador de Desvios de Obstáculos
    // e Medições Laser
    int dist_min, dist_90g, dist_obst = 500; //1000;
    int medidas[8];

    bool exist_obst, exist_obst_ant = false;

    int cont = 0;

    destino.setX(3000);
    destino.setY(-3000);

    ////////// Programa Principal //////////
    do
    {
        desttmp = destino;
        gamma = 0.0;

        posicao = robot.getPose(); // Posição atual do robô
        rho = (double)posicao.findDistanceTo(destino)/1000.0; // Erro de posição do robô [m]
        phi = robot.getTh(); // [graus] orientação do robô
        theta = posicao.findAngleTo(destino); // [graus]
        alpha = theta - phi; // [graus] erro de orientação do robô com relação ao ponto destino

        for (int j = 0; j < 8; j++)
            medidas[j] = robot.getSonarRange(j); //Distância al objeto medida em mm por el sensor i

        // Iniciando verificação para analisar a presença de obstáculos
        dist_min = 3000;
        beta = find_beta(medidas, dist_min, dist_90g);
        exist_obst = find_gamma(dist_min, dist_90g, dist_obst, beta, (double)alpha, gamma);

        if ((exist_obst == true) && (rho > (dist_min * 1.2/1000.0)))
            // Executa desvio tangencial
            tan_escape(robot, posicao, destino, desttmp, beta, gamma, dist_min, medidas);

        // Busca do Alvo real
        // Controlador de Posição
        pos_ctrl(robot, posicao, destino, desttmp, gamma, arq_action_ctrl);

        cont++;

        rho = (posicao.findDistanceTo(destino))/1000.0; // Erro de posição do robô [m]

        ArUtil::sleep(100);
    }while(rho > 0.030 && cont < 1500); // erro maior q 30 mm

    robot.disconnect();
    Aria::shutdown();
    return 0;
}

```